Commodore

COMPUTER CLUB

31 L. 3.500

La rivista degli utenti di sistemi Commodore

Mensile - 25 Maggio 1986 - Anno V - N. 31 - Sped. Abb. Post. GR. III/70 - CR - Distr: MePe

Linguaggio macchina: il tuo sistema personale

Inserto speciale: evitiamo gli errori

Sprotezione d'una cartuccia

Speed DOS: velocizzare il drive

Servizi e giochi per Vic 20, C16, 64, 128

Un segreto del 128

6 7 8

PUR GRN BLU YEL

Y U

H J

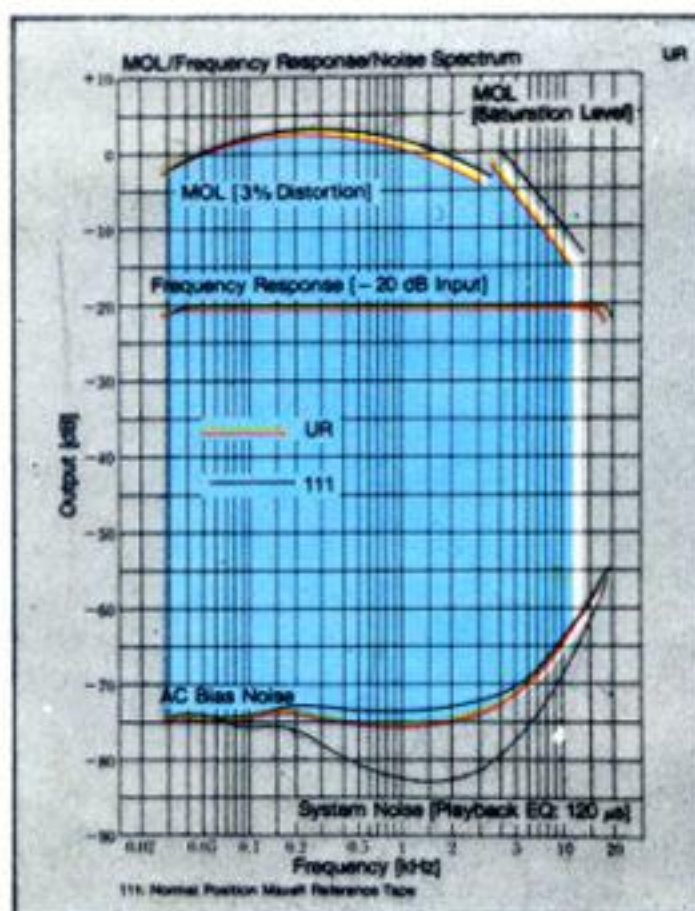
NUOVA MAXELL UR

Una cassetta Low Noise a livello Hi-Fi

C'era una volta la cassetta adatta ad un certo tipo di utilizzo, alla quale non si potevano chiedere prestazioni superiori.

Adesso c'è la UR MAXELL che, grazie alla sua modernissima tecnologia, rivoluziona gli standards della cassetta low noise portandoli a livelli hi-fi.

UR significa UNIVERSAL RECORDING, cioè adatta ad ogni tipo di registratore, dal portatile alla piastra ultrasofisticata, sempre con la certezza di prestazioni eccellenti.

Il nastro UR offre un aumento in MOL (Maximum Output Level) di

1,5 dB alle basse frequenze e di 2 dB alle alte rispetto alla serie UL.

L'ulteriore abbassamento del rumore di fondo offre la gamma dinamica più ampia della categoria: fino a 77 dB (a 1 Khz), ottenendo così un suono chiaro e cristallino senza alcuna distorsione.

L'involucro della UR è costruito con un nuovo polistirene di grande resistenza che ne assicura una lunga vita senza problemi.

A voi non resta che provare; non resisterete al fascino, neanche troppo discreto, della nuova UR MAXELL.

maxell®

L'arte di registrare.

DELPI

Sommario

SPECIALE

E NON INDURCI IN...SYNTAX ERROR

RUBRICHE

Direttore: Alessandro de Simone
Redazione/collaboratori: Claudio Baiocchi, Carlo e Lorenzo Barazzetta, Giovanni Bellù, Simone Bettola, Andrea e Alberto Boriani, Diego e Federico Canetta, Giancarlo Castanga, Umberto Colapicchioni, Pasquale D'Andreti, Maurizio Dell'Abate, Marco De Martino, Piero Dell'Orte, Luca Galluzzi, Michele Maggi, Giancarlo Mariani, Marco Miotti, Flavio Molinari, Claudio Mueller, Enrico Scelsa, D. Matturro, M L. Nitti, Massimo Pollutri, Carla Rampi, Fabio Sorgato, Giovanni Verrelli, Antonio Visconti.
Segreteria di redazione: Maura Ceccaroli, Piera Perin
Ufficio Grafico: Mary Benvenuto, Arturo Ciaglia, Paolo Vertuccio
Direzione, redazione, pubblicità: V.le Famagosta, 75 - 20142 Milano - Tel. 02/8467348
Pubblicità: Milano: Leandro Nencioni (direttore vendite), Giorgio Ruffoni, Roberto Sghirinzetti (settore informatica), Claudio Tidone - V.le Famagosta, 75 - 20142 Milano - Tel. 02/8467348
● Emilia Romagna: Spazio E - P.zza Roosevel, 4 - 40123 Bologna - Tel. 051/236979
● Toscana, Marche, Umbria: Mercurio Srl - via Rodari, 9 - San Giovanni Valdarno (Ar) - Tel. 055/947444
● Lazio, Campania: Spazio Nuovo - via P. Foscari 70 - 00139 Roma - Tel. 06/8109679
Segretaria: Lilliana Degiorgi - **Abbonamenti:** Marina Vantini
Tariffe: prezzo per copia L. 3.500. Abbonamento annuo (11 fascicoli) L. 35.000. Estero: il doppio. Abbonamento cumulativo alle riviste Computer e Commodore Computer Club L. 70.000.
I versamenti vanno indirizzati a: Systems Editoriale Srl mediante assegno bancario o utilizzando il c/c postale n. 37952207
Composizioni: Systems Editoriale Srl - **Fotolito:** Systems Editoriale Srl
Stampa: La Litografica S.r.l. - Busto Arsizio (VA)
Registrazione: Tribunale di Milano n. 370 del 2/10/82 - Direttore Responsabile: Michele Di Pisa
Sped. in abb. post. gr. III - Pubblicità inferiore al 70% - **Distrib:** MePe, via G. Carcano 32 - Milano

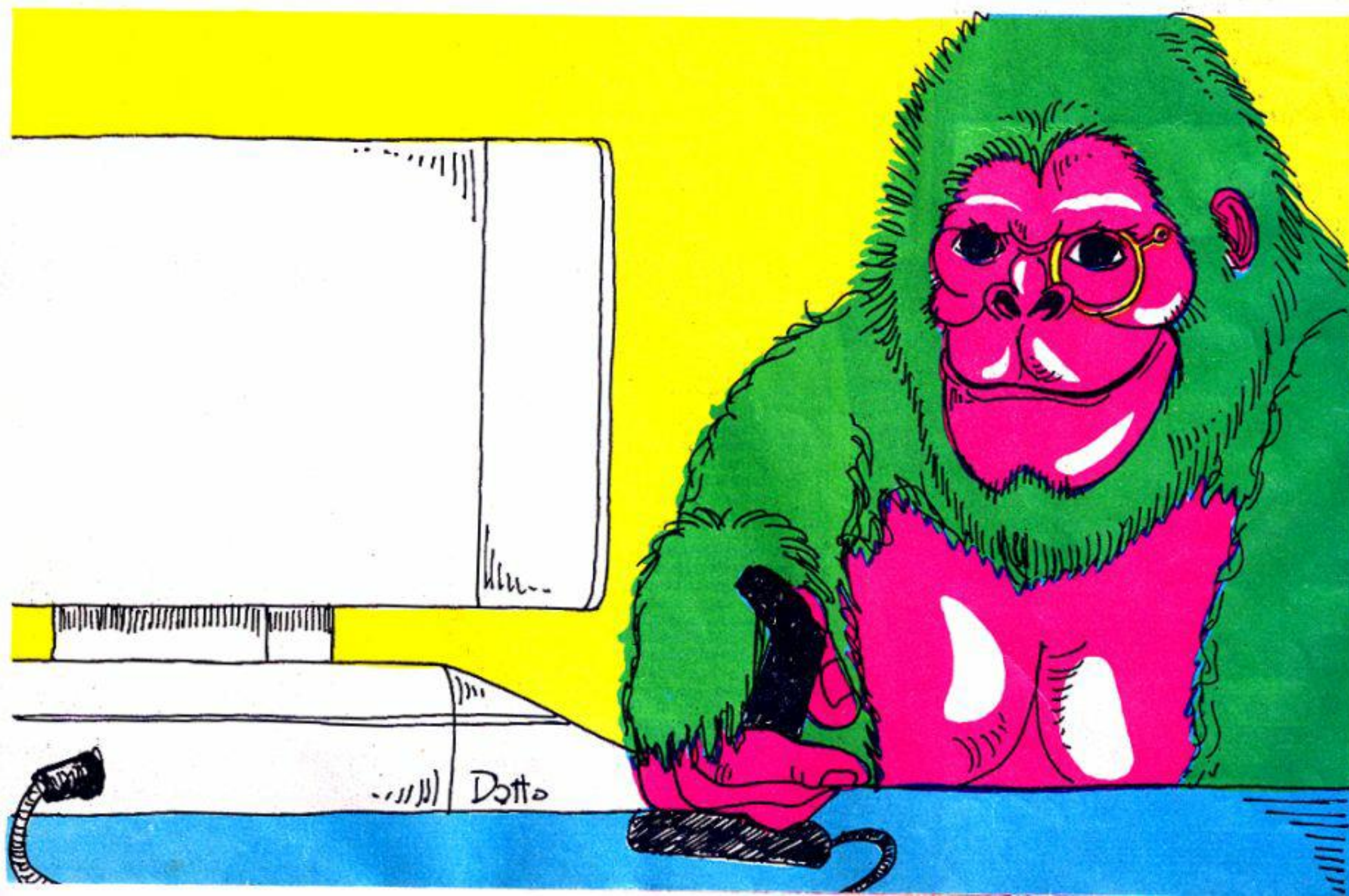
Stupido come un computer

Che influenza può avere un calcolatore su chi lo usa? E' vero che, a furia di usarlo, si diventa aridi e privi di creatività? Ed altre dissertazioni...

di Alessandro de Simone

Come è ormai consuetudine da qualche tempo, anche questo mese viene presentato su CCC, in forma "speciale", un argomento di interesse generale. Stavolta ci occupiamo di come interpretare correttamente gli errori da noi commessi in fase di stesura di un programma Basic.

Benchè il termine sia sproporzionato alle funzioni svolte da un calcolatore in fase di "debug", non è azzardato affermare che le segnalazioni di errore costituiscono un esempio di intelligenza artificiale (A.I.)

E' ben vero che il messaggio Syntax Error, tanto per citare il più banale, non

è altro che il risultato di un mancato riscontro con parole chiave (o altro) memorizzate nel computer. Ma allora, volendo privare la visualizzazione del messaggio della caratteristica di "intelligenza", cosa dovrebbero rappresentare i segni rossi e blu apposti da un insegnante nella correzione di un compito di grammatica, di sintassi o di matematica?

Fortunatamente, a scuola, il corpo docente non si limita a segnalare l'errore commesso ma (almeno così dovrebbe essere) aggiunge uno o più commenti scritti, per non parlar poi delle ramanzine orali...

Ma ciò che interessa in questa sede non è lo stabilire se un computer può pensare o se è soltanto uno "stupido" al nostro servizio (fatto che, a mio modesto e personale parere, può esser facilmente assimilato ad una forma mascherata di razzismo o neo-classismo).

Vogliamo qui individuare, nello strumento informatico, gli atteggiamenti che la mente umana è costretta (o desidera) assumere nell'evoluzione della società.

L'altro ieri

Tanti anni fa c'era, tra le altre discipline scolastiche (che allora si chiamavano, brutalmente, "materie") quella della Calligrafia che era tenuta in gran conto dall'intero corpo docente nella valutazione (ma si diceva, allora, "voti") dei risultati finali. Erano posti in commercio, addirittura, speciali quaderni con righi strani prestampati, simili a spartiti musicali, che facilitavano la "bella scrittura"; parallelamente fioriva il mercato dei pennini, con vari "becchi", ciascuno adatto per specifiche applicazioni.

Fortunatamente la diffusione della penna a sfera prima, e del buon senso poi, limitò la proliferazione dell'inchiostro con gran dispiacere delle fabbriche di detersivi. Al giorno d'oggi, comunque, poderose masse di impiegati statali sono costrette, da apposite leggi (o, più semplicemente, da stantia imbecillaggine) ad usare la penna invece dei computer che sono sospettati, come i comunisti degli anni '50, di mangiare i bambini.

Lasciamo però da parte la mancata meccanizzazione delle informazioni e, con essa, il conseguente accumulo di pratiche che, tra un po' di tempo, richiederanno gli stadi comunali per esser archiviate.

Ieri

Ricordo che, iscritto al biennio di ingegneria, chiesi ad un professore se quella matematica che ero costretto a studiare (limiti, derivate, integrali e compagnia) sarebbe stata utile per progettare un edificio o per determinare il carico dell'acciaio. L'interpellato, candidamente, rispose che per far reggere in piedi un edificio, uno stadio gremito o una nave stracolma sono sufficienti calcoli che richiedono le moltiplicazioni, le divisioni e (ma raramente), anche le somme e le sottrazioni. Di logaritmi, limiti, derivate e sciagurati compagni neanche l'ombra, almeno nella più comune pratica professionale.

In seguito, passato al tiennio superiore, e applicandomi nelle esercitazioni pratiche (progetto di edifici in cemento armato, telai in acciaio e applicazioni del precompresso) scoprii che il professore di qualche anno addietro aveva perfettamente ragione: erano sufficienti le quattro operazioni.

Ma allora perchè costringere gli studenti ad affrontare argomenti così odiosi che suggerivano, molto spesso, il cambio del corso di studi?

La risposta, in verità, mi fu fornita dallo stesso insegnante: "Perchè l'analisi matematica è l'unico strumento che attualmente abbiamo per conferire una tipica forma mentis al futuro ingegnere che dovrà barcamenarsi tra situazioni sempre nuove e impreviste, mutevoli da progetto a progetto". E anche in questo aveva perfettamente ragione.

Oggi

Ma i tempi son cambiati. Prima degli anni '70 i calcolatori erano ancora una chimera e il vecchio CDG 20 (a transistor!) installato presso la facoltà di Ingegneria di Napoli, funzionava un giorno sì e tre no, con gran disperazione del giovane e valido professor Esposito al quale, comunque, riconosco il merito del contagio per la passione dei computer.

E, oggi, quale strumento "logico" migliore, se non un calcolatore, può esser proposto a chi vuole esplorare, anche da solo, i meandri più nascosti della razionalità? Quale miglior stimolo ad applicarsi può esser rappresentato dal puzzle di un messaggio di errore inspiegabile? Oppure di un risultato decisamente contrario a quello che ci si aspettava?

Facciamo, tutti, un passo indietro nel considerare le nostre prime esperienze con un calcolatore. Se qualcosa non funzionava (ed eran tante, specialmente all'inizio) pensavamo subito che il computer acquistato era rotto dato che (inevitabile, presuntuosa premessa) noi non potevamo aver sbagliato.

Chi, invece, ha trascorso centinaia di ore alla tastiera, ma non per giocare, sa benissimo che, in caso di malfunzionamenti, la causa principale siamo noi stessi, che non siamo stati capaci di programmare o di trascrivere un listato oppure di non aver sistemato alcuni puntatori prima di o dopo che.

Una nuova umiltà, unita ad una buona dose di pazienza, caratterizza l'umore dell'"Homo programmans". Miglioramenti nel rendimento scolastico, maggior capacità di effettuare scelte oculate, razionalità, curiosità, desiderio di conoscenza e rifiuto di comportamenti contraddittori costituiscono il risultato di intense ore di (piacevole) studio di fronte al monitor. Per non parlar, poi, degli appassionati di linguaggio macchina.

E pensare che c'è ancora gente, sostenuta da scellerati tuttologi, che vede il computer come una nuova peste da allontanare con diffidenza.

Il tutto in un contorno sociale spesso irrazionale e contraddittorio che, anzi, richiede carenza di razionalità per sopravvivere.

Scherzi

Come tutti, o quasi, hanno capito dalla data e dal periodo cui si riferiva (:carnevale), il "Comunicato" apparso sul numero scorso era un pesce d'aprile.

Ma, a proposito di scherzi, siamo costretti a segnalare che alcuni buontemponi, non sappiamo se in buona fede oppure no, si presentano a nostro nome per compiere azioni che, superando la barriera del buon gusto, cadono nella illegalità.

Ci riferiamo agli inviti, da parte di persone non identificate, a partecipare a concorsi inesistenti, al ritiro di articoli e programmi con promessa di pubblicazione o di commercializzazione che mai sono da noi state autorizzate.

Pertanto non solo invitiamo gli autori delle "bravate" a interrompere tali azioni ma, nel contempo, consigliamo i nostri lettori di diffidare di persone che dovessero presentarsi a nostro nome per ritirare denaro, articoli o programmi.

In caso di dubbio telefonate (02/8467348) per segnalare sospetti di irregolarità e, nel frattempo, non consegnate nulla a nessuno. Tutti i componenti della Redazione sono, ovviamente, sempre informati sulle varie iniziative promozionali in corso e, pertanto, sono in grado di sciogliere ogni dubbio.

Pulsante inefficace

□ **Ho costruito, come da voi suggerito sul N.28, il pulsante di Reset, ma non funziona. Potete aiutarmi?** *(Massimo Musacchi, Ferrara)*

• Purtroppo, come avevamo specificato nello stesso articolo, l'esemplare di C/64 in tuo possesso non è di quelli che consentono tale applicazione. In questi casi è necessario utilizzare la porta utente oppure quella di espansione per "ricavare" il pin di reset. Se ritieni che la spesa dei connettori per tali porte sia eccessiva, l'unico modo di aggirare l'ostacolo è quello di aprire il computer, rintracciare il pin di Reset sul circuito stampato (meglio se nei pressi delle porte) ed effettaure il collegamento in loco. Attento a non creare involontari corti-circuiti col saldatore!

Dove lo metto?

□ **Ho costruito il pulsante di Reset ma, quando collego il drive, devo necessariamente rimuoverlo. Come posso continuare ad utilizzarlo?** *(Amerigo Santoro, Terni)*

• Il pulsante di Reset collega a massa il pin dell'intero sistema (computer e accessori). E' quindi possibile inserirlo nella seconda porta seriale del drive. E se a questa è collegata la stampante, inseriscilo nella porta libera di quest'ultima periferica.

Joy e paddle

□ **In un gioco in cui è previsto l'uso del joystick, è possibile usare le paddle (e viceversa)?** *(Mario Criscuolo, Roma)*

• Un joystick è formato da un gruppo di cinque interruttori che, a seconda della posizione dell'assicella, vengono chiusi oppure aperti e, in tal modo, il computer "riconosce" lo spostamento impresso.

Il joystick, pertanto, è uno strumento del tipo "tutto o niente" nel senso che un interruttore può esser solo aperto oppure chiuso, e non sono possibili "stati" intermedi.

Le paddle, al contrario, sono resistenze elettriche variabili che, collegate a porte "analogiche", comunicano al computer una delle 256 possibili (almeno in teoria) posizioni intermedie della manopola.

Pertanto non è possibile la sostituzione accennata sia perchè il joy è digitale e le paddle sono analogiche, sia perchè le paddle sono due e il joy possiede al suo interno ben cinque interruttori sia perchè, infine, i pin di collegamento presenti nel connettore sono diversi nei due casi!

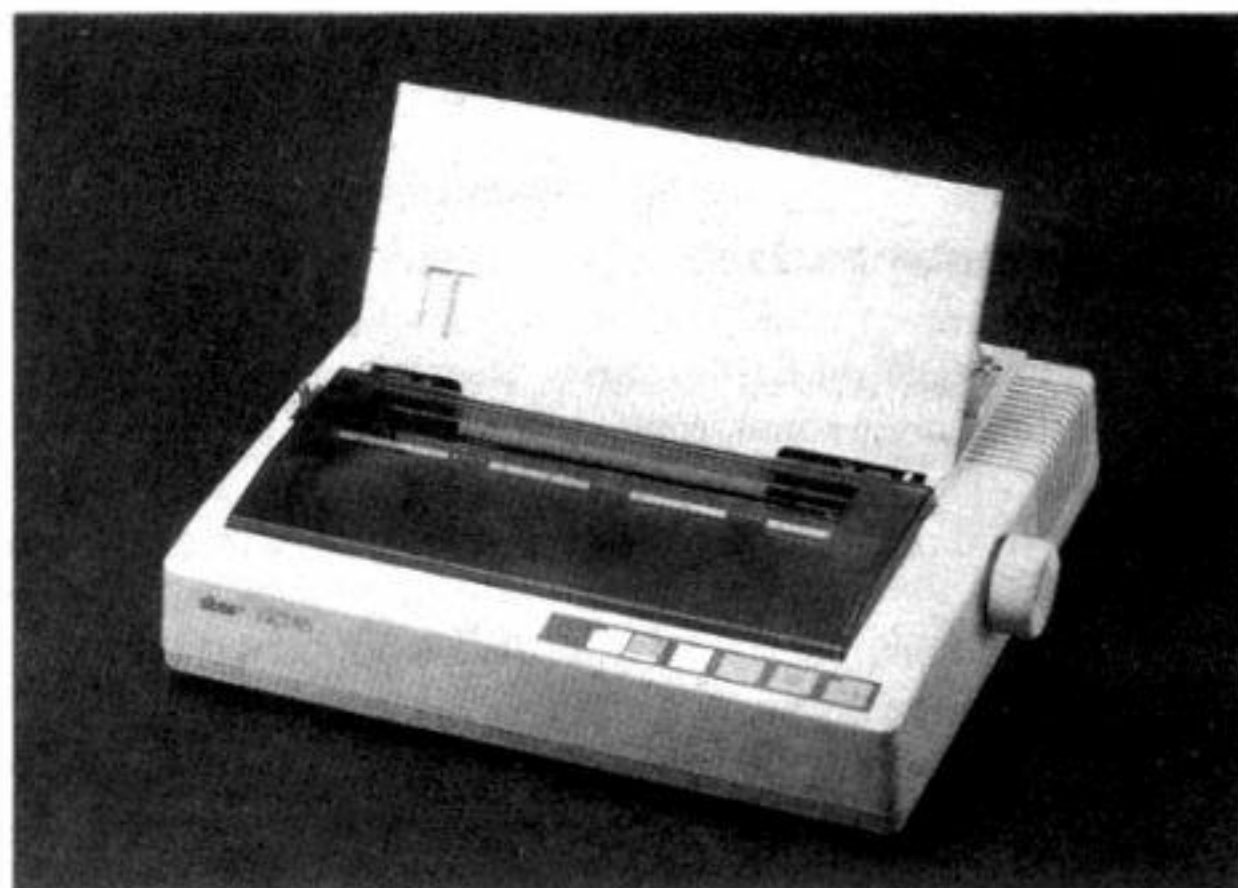
Mancanza di "Press Play"

□ **Da alcuni mesi il mio computer, al comando Load e Save, non risponde più col solito messaggio "Press..." ma esegue immediatamente, pur funzionando benissimo. Come posso fare per evitare questo difetto che è davvero seccante in programmi che richiedono esplicitamente la pressione del tasto Stop del registratore?** *(Giovanni Ripamonti, Lecco)*

• Abbiamo più volte detto che, se il computer funziona egualmente nelle varie fasi (registrazione, verifica, caricamento) il difetto è da attribuirsi ad un corto circuito interpretato, dal computer, come un costante abbassamento di un tasto del registratore.

Tale corto circuito può essersi verificato sul computer (caso raro) oppure sul registratore (più frequente).

Per sapere quale dei due casi è il tuo, stacca il registratore e digita "Load". Se lo schermo "scompare" subito vuol dire che in corto è sicuramente la pista del

La Superstar fra le stampanti per computer é una Star!

Probabilmente, nessun'altra stampante riunisce in sè tutte le straordinarie prerogative della **NL-10**, una periferica per computer estremamente convincente nelle prestazioni e nel prezzo. **NL-10** può contare su fans in ogni settore aperto all'informatica: gestionale, organizzativo, amministrativo, sviluppo, produzione, hobbystico. Di lei gli addetti ai lavori apprezzano la semplicità d'uso e la qualità dello stampato. E' sorprendente su **NL-10** la quantità di funzioni di stampa, controllabili dall'utente tramite un pannello frontale molto sofisticato, cosi come la varietà dei formati di stampa e la sua enorme adattabilità a qualsiasi tipo di computer. Anche nell'affidabilità, **NL-10** darà prova di tutta la sua amicizia. Chieda al nostro rivenditore di zona una dimostrazione di Superstar **NL-10**: siamo certi che anche Lei concluderà che, **con una Star, si può andare molto lontano!**

La tua stampante

Via Gallarate, 211 20151 Milano
tel. 02/301.00.81 r.a. 301.00.91 r.a.

Per avere maggiori informazioni e l'indirizzo del rivenditore della Sua zona, ci invii il coupon allegato.

Ditta: ______ Via: ______ n°: ______

Nome: ______ Cap.: ______ Città: ______

______ Tel.: ______

calcolatore. Se, invece, compare la scritta "Press..." vuol dire che l'inconveniente è da ricercarsi nel registratore.

In entrambi i casi è necessaria un po' di pratica sia per individuare il corto circuito sia per eliminarlo. Se è dovuto a cause meccaniche, un metodo sbigativo è quello di troncare il filo del sensore di tasto premuto e di collegarlo ad un pulsante esterno (normalmente aperto) che premerai quando richiesto.

Computer parlanti

☐ **Posseggo un C/16 e vorrei sapere se esistono programmi, come il vostro "La voce" per C/64, capaci di farlo "parlare"** *(Franz Comploj, Ortisei)*

• La sintesi vocale, e altre diavolerie elettroniche capaci di simili meraviglie, presuppongono l'impiego di sofisticati circuiti integrati appositamente realizzati. Il SID del C/64 è attualmente il miglior sintetizzatore sonoro attualmente installato in computer di basso costo. Quello montato sul C/16, purtroppo, possiede caratteristiche nettamente inferiori e non sarà mai possibile, di conseguenza, far parlare il tuo calcolatore.

Koala non funzionante

☐ **Non riesco a far funzionare il vostro programma "Koala video" pubblicato nell'enciclopedia di routine (CCC N.27).** *(Marco Coccia, S.Teresa di Spoltore)*

• Il programma cui ti riferisci consente di utilizzare, nei propri listati, disegni realizzati in precedenza col sistema software - hardware "Koala" che, tra l'altro, funziona solo col drive per dischi 1541. Purtroppo tu possiedi solo il registratore e il programma non può funzionare.

C/64 a 80 colonne

☐ **Vi sono in commercio alcuni programmi per C/64 che consentono di portare lo schermo da 40 a 80 colonne. Di che cosa si tratta?** *(Marco Russo, Siracusa)*

• Si tratta di utility che ricorrono alla tecnica della ridefinizione dei caratteri per fare in modo che, nello stesso spazio di una cella video in cui normalmente risiede un solo carattere, ve ne siano due. Come puoi immaginare la nitidezza decade in modo impressionante, la possibilità di gestire i colori rimane illusoria ed è praticamente impossibile riconoscere i singoli caratteri su un comune televisore, sia pure in B/N. Non dimenticare, inoltre, che programmi professionali (come word processor, data base, spreadsheet ed altri) non appena vengono caricati "cancellano" tale utility e lo schermo ritorna alle consuete 40 colonne.

Videoregistrare le schermate

☐ **E' possibile collegare il videoregistratote al computer?** *(Davide Militello, Limbiate)*

• Puoi collegare il cavetto che normalmente inserisci nella presa del T/V, nell'apposita presa d'ingresso del videoregistratore. Tutto qui.

Caldi comportamenti

☐ **Dopo mezz'ora circa di lavoro il mio computer "impazzisce". A che cosa può esser dovuto questo strano comportamento a "caldo"?** *(Claudio Della Pietra, Abano)*

• Abbiamo più volte affermato che i malfunzionamenti che si verificano quando il computer si riscalda sono dovuti a saldature "fredde" che, a temperatura ambiente, consentono il corretto passaggio della corrente. Quando il computer (drive, stampante o altro) si riscalda, le dilatazioni termiche, benchè minime, interrompono o, comunque, alterano il normale flusso di energia. In questi casi, se si è esperti, è sufficiente aprire il computer e "ripassare" le saldature sospette. Se non si è pratici di saldature è meglio inviare l'apparecchio ad un centro specializzato in riparazioni di computer.

Sfortunato Plus/4

☐ **E' possibile trovare programmi gestionali per il Plus/4 oppure adattare facilmente programmi normalmente in uso per il C/64?** *(numerosi lettori).*

• No

Fine di un computer

☐ **E' vero che la Commodore ha deciso di bloccare la produzione del C/64? E, in caso affermativo, che danno potremmo avere noi, utenti di questo computer?**

• Le voci che si sentono in giro sono di vario tipo: Gheddafi proposto per il Nobel della pace, Reagan che trascorre le vacanze a Tripoli, Gorbaciov che propone il divieto della vendita perfino delle armi giocattolo. E' inevitabile che corrano anche voci di fallimento della Commodore, di fusione della IBM con la Sinclair e di altre deliziose notizie.

Qualunque sia la salute commerciale della Commodore è impensabile, a nostro parere, che un modello di computer, simbolo della stessa Commodore, scompaia dopo esser entrato nella leggenda.

Comunque vadano le cose stà tranquillo: il numero di utenti è troppo alto perchè software house, fabbriche di accessori e case editrici del settore (tra cui noi, in prima fila) smettano di occuparsi del popolarissimo calcolatore. Pensa che ancora oggi, a distanza di anni dal fallito esperimento della Texas, continuano a circolare programmi, periferiche e software per il modello TI/99 che ebbe una diffusione, ed un successo, nettamente inferiori al C/64!

Banche dati

☐ **Potete pubblicare i numeri di telefono**

di alcune banche dati operanti in Europa?

• Sul N.2 (febb/mar '86) della nostra rivista "Personal Computer" è stato affrontato il tema delle modalità di collegamento via modem. Nello stesso articolo (Ulisse a ITAPAC) figura anche un elenco, piuttosto ampio, di banche dati e relativi numeri di telefono.

Tuttoschermo

□ **E' possibile eliminare la cornice del video per utilizzare l'intero schermo?**
(Alfonso Perna, Montella)

• Il bordo del video, presente anche per una questione estetica, consente di adattare qualsiasi esemplare di computer prodotto a qualsiasi televisore esistente.

In assenza del bordo, infatti, alcune colonne o righe di caratteri potrebbero non comparire su certi esemplari di TV con le conseguenze che è facile immaginare. Il manuale di istruzioni del C/16 (e Plus/4) accenna a questa anomalia che potrebbe verificarsi e, addirittura, è previsto un apposito comando (Esc +C) per ridurre le dimensioni dell'area video in modo da esser completamente visibile anche con televisori che dovessero presentare qualche problema.

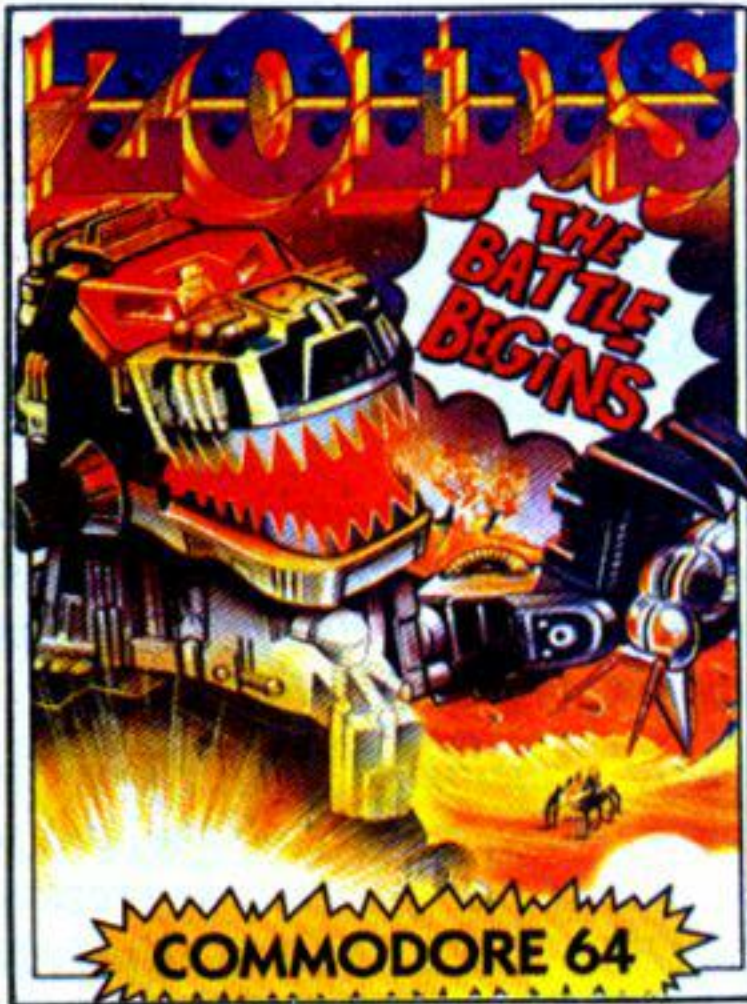
Il Vic 20 aveva la possibilità di eliminare il bordo (o meglio, di allargare le dimensioni dell'area destinata alla visualizzazione dei caratteri) ma tale tecnica non è stata utilizzata spesso in programmi posti in commercio proprio perchè non si aveva la garanzia che l'acquirente, tornato a casa, potesse utilizzare il programma col proprio televisore.

Se può interessare, un apposito programma per il Vic 20 (Scrittura a tutto schermo) che allargava il video a 33 righe per 25 colonne è stato pubblicato su CCC N.11.

AVVERTENZA IMPORTANTE

Tra le numerose domande che pervengono in Redazione, vengono scelte in prevalenza quelle di interesse generale, rinviando, spesso sinedie, la pubblicazione di quelle ritenute meno interessanti oppure già affrontate di recente su Commodore Computer Club.

Invitiamo pertanto i lettori, specie se principianti, a sfogliare con attenzione le pagine della nostra rivista e a leggere articoli che, pur se apparentemente non li interessano, contengono purtuttavia preziose informazioni su istruzioni particolari (POKE, PEEK, SYS ed altre) o tecniche insolite di programmazione, registrazione di dati, programmi eccetera.

Zoids

Questo nuovo gioco della Mastertronics presenta alcune simpatiche novità. Innanzitutto una musica di sottofondo, ed una schermata visualizzata in permanenza, accompagnano l'intero caricamento del gioco che si compie dopo 92 giri di nastro.

Quando il gioco inizia, la parte centrale del video viene occupata dallo schermo radar, in alta risoluzione, incaricato di visualizzare la posizione della nostra navicella all'interno di un labirinto. E' ovvio che la trama del gioco (combattimenti contro i soliti "cattivi") è solo una scusa per mettere alla prova la nostra abilità nel manovrare il joystick e, finalmente, anche la nostra intelligenza. E' infatti indispensabile, servendosi del joy, non solo governare correttamente il movimento dell'astronave ma, soprattutto, rintracciare e ricomporre i numerosi pezzi di un oggetto precedentemente esploso, prima che se ne impossessino i nemici.

Per raggiungere lo scopo sono costantemente presenti sullo schermo, oltre al radar che consente perfino di zoomare la nostra posizione, altre otto "icone" (porzioni di schermo in alta risoluzione). Queste, toccate dal joy, forniscono varie informazioni oppure, a seconda dei casi, eseguono comandi opportuni per ciò che riguarda combattimenti, tempeste, attacchi nemici, esplorazioni di città o attivazione di scudi.

Due forme d'onda, rappresentanti la pulsazione del nostro cuore e quella Zoid (che si riferisce ai danni subiti) completano le informazioni necessarie a condurre con successo il gioco che rappresenta un felice compromesso tra un classico game d'azione e il genere "adventure".

Game Killer

Questa cartuccia della Mastertronic rappresenta senz'altro una novità nel campo dei videogiochi.

Inserita, a computer spento, sul retro del C/64, consente di caricare, da nastro oppure disco, un qualsiasi videogioco e di giocarlo come al solito con un'unica sostanziale differenza: è possibile eliminare il riconoscimento delle collisioni sprite-sprite, quelle sprite-caratteri oppure qualsiasi collisione.

Il vantaggio di un tale provvedimento è evidente: poichè la maggior parte dei giochi utilizza la collisione degli sprite per incrementare le penalità, in questo modo il gioco non "si accorge" dell'avvenuta collisione e il numero di "vite" rimane sempre lo stesso!

E' ovviamente possibile ripristinare le condizioni normali del gioco in qualsiasi momento premendo il tasto incorporato nella cartuccia.

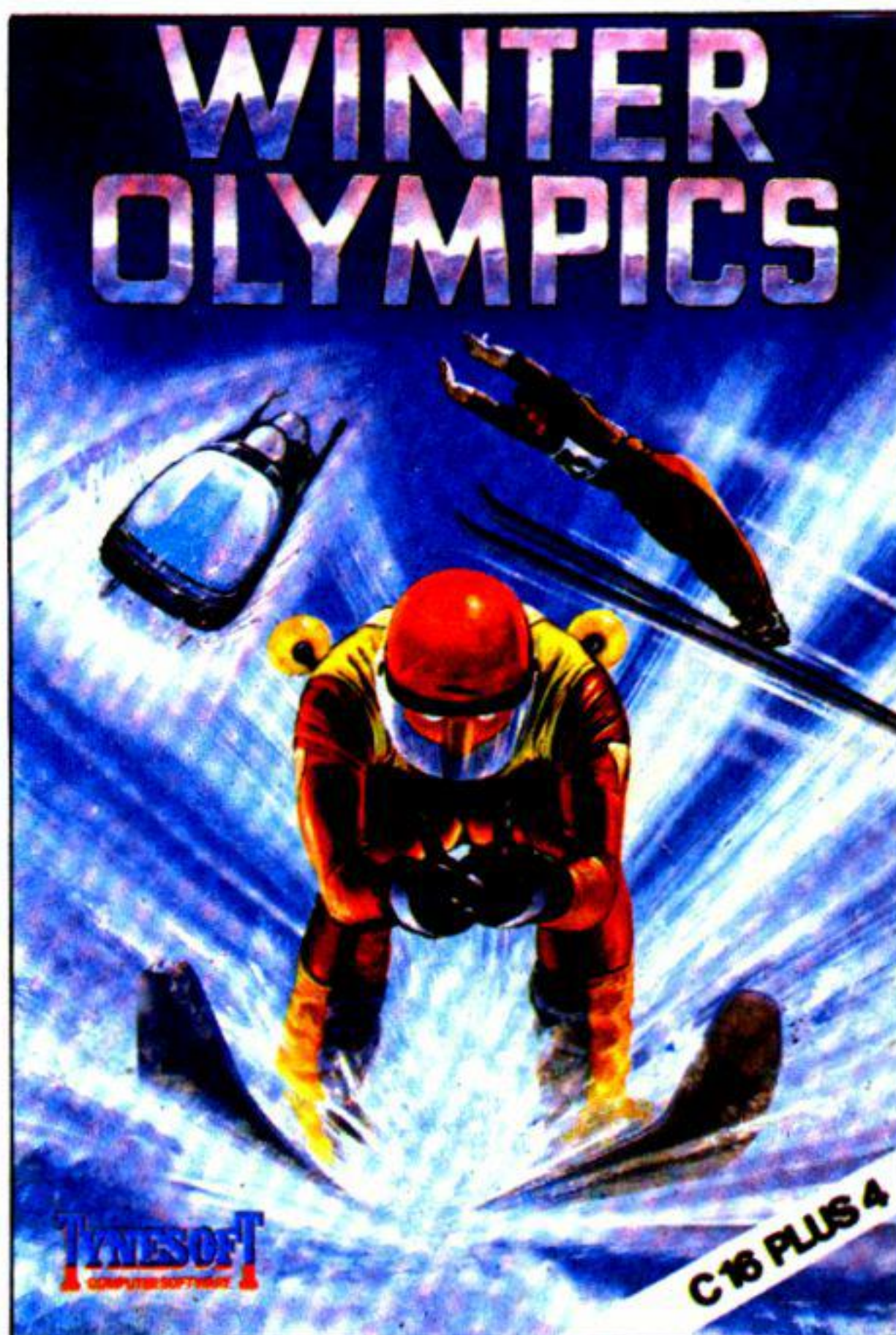
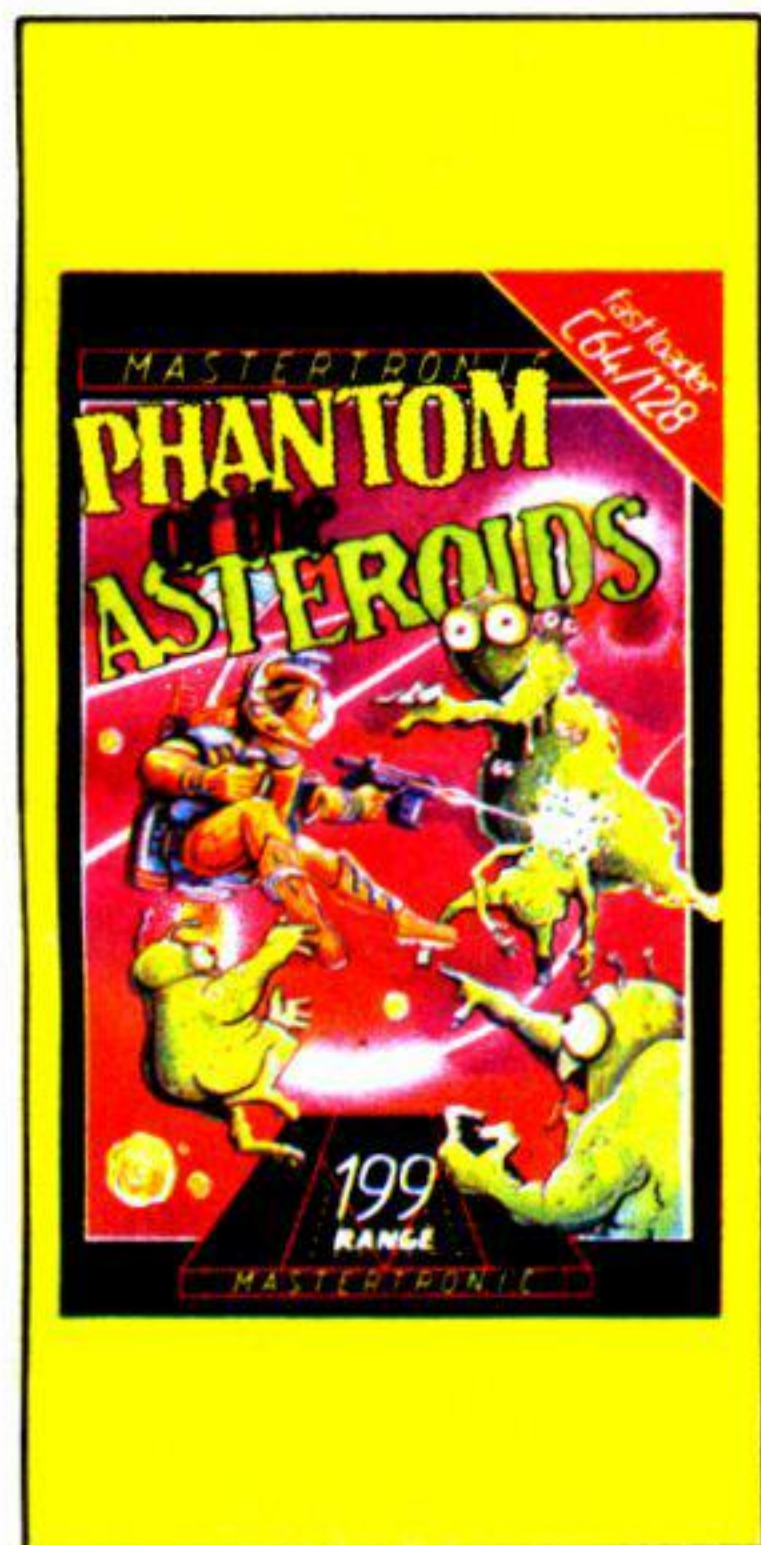
Game Killer risulta utilissimo per vedere, finalmente, le schermate mai giocate di videogiochi quasi impossibili. Dopo aver effettuato alcune prove, ci sembra però doveroso precisare che non in tutti i videogiochi la cartuccia ha funzionato.

In alcuni game, inoltre, dato che viene disabilitato il riconoscimento di qualsiasi collisione, è necessario ricorrere spesso alla continua abilitazione e disabilitazione della cartuccia in modo da continuare a "vivere" quando il nemico ci colpisce e, al contrario, di manifestare la nostra presenza nei casi in cui è necessario "toccare" particolari oggetti oppure colpire il nemico (sprite) con nostre bombe e proiettili (ancora sprite) per totalizzare punteggi a nostro vantaggio.

Commodore 64

Phantom of the asteroids

Un ennesimo gioco in cui è necessario colpire i nemici evitando di esser toccati o di entrare in collisione con raggi laser o spigoli del labirinto. Simpatica la musichetta di accompagnamento e abbastanza realistico il movimento dell'omino quando si solleva grazie al reattore del suo zaino.

Ideale per chi non è ancora stanco dei classici videogame che richiedono molta destrezza ed abilità nel manovrare il joystick.

C/16 Winter Olympics

Avevamo affermato, tempo fa, che un videogioco per C/16 non avrebbe mai potuto competere con analoghi videogame per C/64 perchè quest'ultimo è dotato di sprite dalle possibilità impareggiabili.

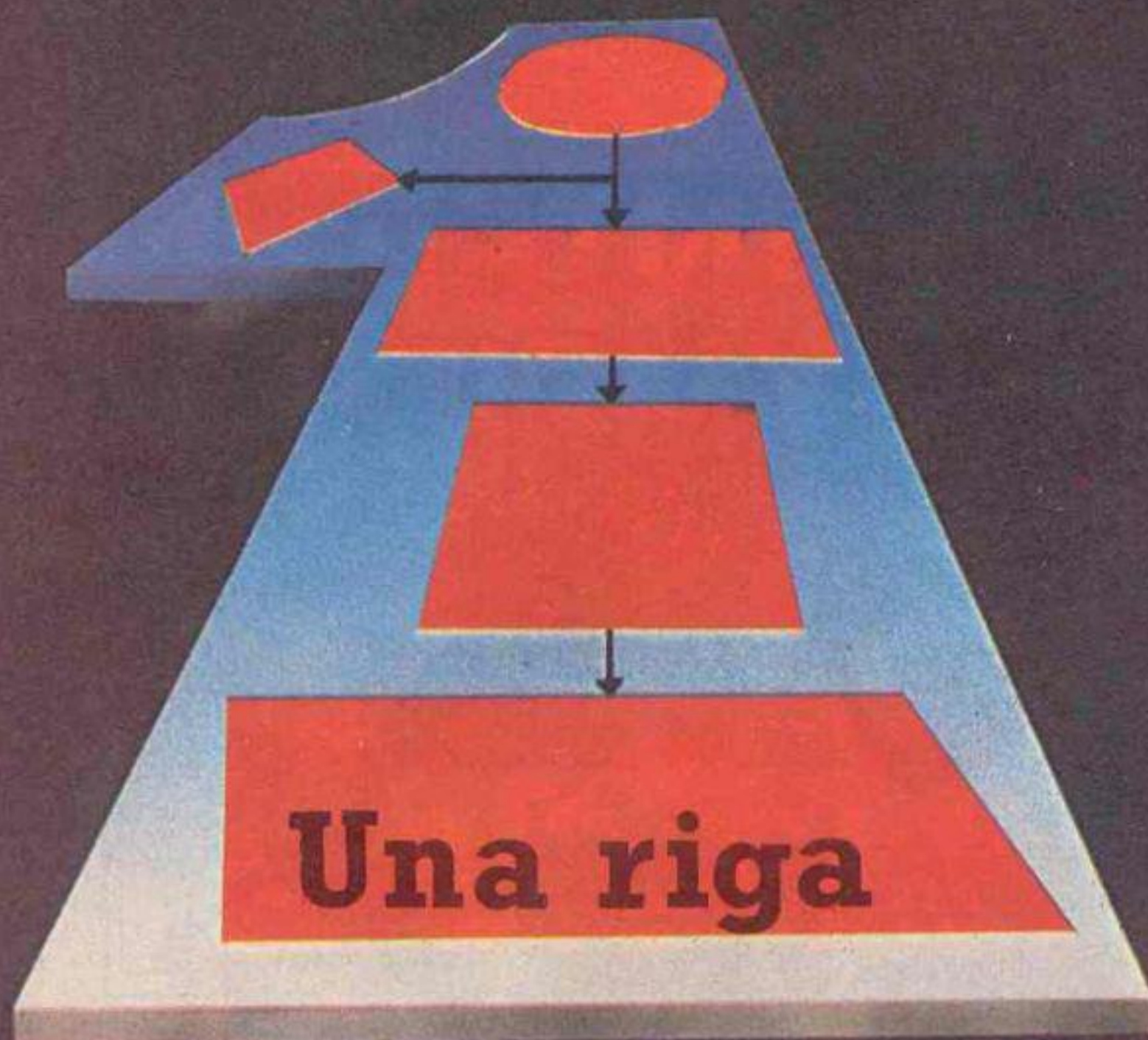
Questo nuovo gioco della Mastertronic, invece, ci smentisce e ne siamo felici.

In effetti non è altro che una copia più o meno fedele del popolarissimo Winter Games (per C/64) dotato di un'animazione che sicuramente sfrutta al massimo le capacità grafiche del piccolo computer.

Vi sono ben sei tipi di gare da caricare una alla volta.

Speed Skating, che è una gara di velocità sui pattini contro il computer; Ski Jump, classica gara di salto dal trampolino; Cresta Run, simulazione piuttosto realistica della nota competizione su Bob; Ski Slalom, che non necessita di spiegazioni; Curling, gara di bowlig su ghiaccio; Down Hill, simile allo Slalom ma gli oggetti da evitare sono gli alberi e non le bandierine.

Buona la grafica, specie se si dispone di un TV color, e soddisfacente l'animazione che sembra quasi realizzata, come dicevamo prima, ricorrendo alla tecnica degli sprite.

Una riga

Su ogni numero di Commodore Computer Club compaiono un paio di pagine dedicate a una dozzina di micro programmi lunghi una sola riga.

E' ovvio che non è possibile pretendere effetti sorprendenti in listati così brevi, tuttavia, sviluppando l'idea su cui gli stessi programmi son basati è possibile pervenire a realizzazioni di tutto rispetto ricorrendo, è inutile dirlo, a qualche riga in più.

Particolarmente interessanti per i principianti, spesso utili anche per gli esperti, i micro programmi di una sola riga rappresentano una valida "palestra" per abituarsi a concentrare in poche, essenziali istruzioni la soluzione di problemi complessi solo in apparenza.

Avvertenza:
Come specificato altrove (nel riquadro di queste pagine), alcune righe occupano più di 80 caratteri. Nulla vieta, salvo rarissime eccezioni, di digitare gli stessi comandi in due o più righe Basic numerate in successione. Eviterete, in tal modo, di ricorrere alle abbreviazioni e di modificare con maggior facilità i comandi eventualmente digitati in modo errato.

Listati per il solo C/128

1

Errori. Tutti i messaggi di errore del C/128 verranno visualizzati grazie a questa riga.

```
1 FORI=1TO41:
  PRINTERR$(I):NEXT
```

2

Typewriter. Il vostro computer trasformato in una macchina da scrivere con tanto di campanellino per ogni tasto premuto.

```
1 PRINTCHR$(164)CHR$
  (157);:GETKEYA$:
  PRINTCHR$(7)A$;:GOTO1
```

3

GO64 con SYS. Lo sapevate che esiste un altro modo per passare, da C/128, in modo C/64? Eccolo:

```
1 SYS65357
```

4

Ridefinizione Help. Anche il tasto HELP può essere ridefinito: provare per credere...

```
1 DATA  71,79,54,52
 ,13:FOR I=0 TO 4:
 READ A:POKE 4168+
 I,A:NEXT
```

5

Disegni musicali. Un disegno accompagnato dal campanellino ottenuto sia con CHR$(7) che con Control "G".

```
1 GRAPHIC1,1:
  FORI=1TO180STEP5:
  BOX1,70,100,220,60,I
  :PRINTCHR$(7):NEXT:
  GETKEYA$:GRAPHIC0
```

6

Reset. La SYS di reset del C/128 (usatelo anche per il BOOT).

```
1 TEMS=65341:SYSTEMS
```

Per C/64 e C/128

7

Riempischermo Basic 64/128. Una riga per riempire lo schermo con un carattere.

```
1 CLR:
  FORI=1024TO2023:
  POKEI,65:NEXT
```

8

Riempischermo L.M. 64/128. Lo stesso programma di prima scritto, però in Linguaggio Macchina. Visto che velocità?

```
1 DATA 169,65,162,0
 ,157,0,4,157,0,5,
 157,0,6,157,235,6
 ,232,208,241,96:F
 OR I=8192 TO 8211
 :READ A:POKE I,A:
 NEXT:SYS8192
```

Per C/16 e C/128

9

PRINT Monitor. Una carenza dell'utility (incorporata) "Monitor" del C/128 e del C/16 è quella di non avere il comando Print per stampante. Ovviamo all'inconveniente con questo microlistato. Una volta dato il RUN qualsiasi comando impartito provocherà la stampa su carta (es. D 0800).

```
1 OPEN4,4:CMD4:MONITOR
```

Trucchi e trucchetti per C/64

Un gruppo di "una riga" e consigli vari per effetti insoliti. Listati e Poke tutti da provare e utilizzare nei propri programmi, magari per sperimentare nuove forme di protezione...

10

LIST senza numeri. Qualsiasi vostro listato apparirà, ma senza i numeri di linea.

```
1 POKE 22,35:LIST
```

11

Velocità cursore. La locazione 56325 contiene la velocità del cursore: modificandola si ottengono interessanti effetti.

```
1 INPUT "(NORM=48)
```

```
VELOCITA' (1-255)
";V:POKE 56325,V
```

12

Typewriter. Il vostro C'64 non accetterà alcun comando grazie alla semplice modifica della locazione 120.

```
1 POKE 120,0
```

13

LIST di... numeri. Verranno listati solo i numeri di linea. A che cosa può servire? Decidetelo voi!

```
1 POKE 774,0
```

14

Reverse Poke. Un altro modo per ottenere scritte in reverse.

```
1 POKE 199,1:PRINT"
  MODO REVERSE ATTI
  VATO !!"
```

15

Reset colori. Come tutti (!) sanno, il comando SYS 64738 resetta il computer inizializzando anche i colori; tramite questa riga potrete resettarlo ugualmente, mantenendo nel contempo i colori precedentemente settati.

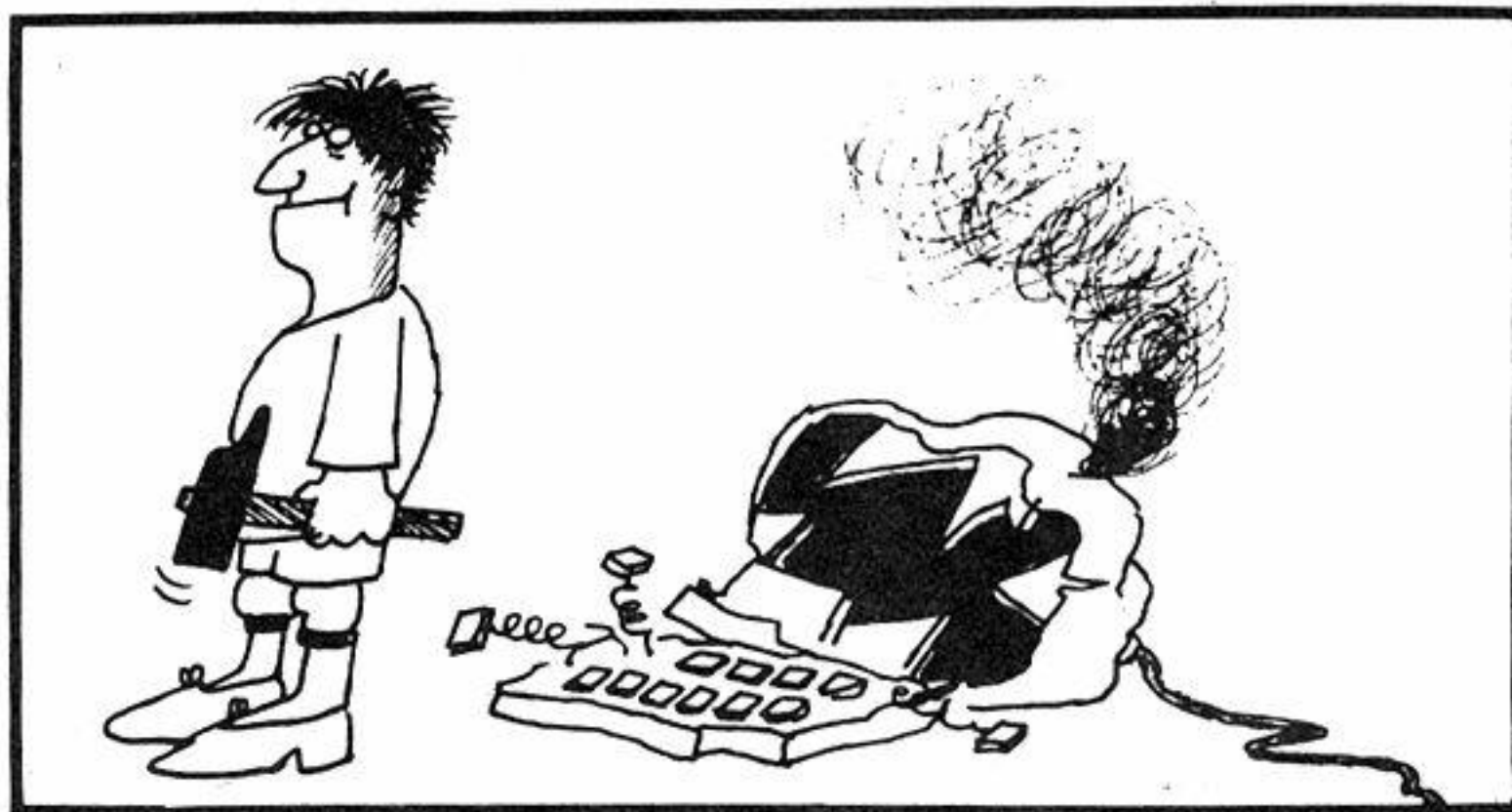
```
1 POKE 53280,1:POKE
   53281,1:SYS 64767
```

Michele Maggi

Due suggerimenti di Mario Criscuolo - Roma

Autorepeat tasti. Con Poke 650,128 si abilita il Repeat automatico per tutti i tasti.

Recupero programma. Con questo gruppo di comandi, da digitare dopo un involontario NEW oppure un'accidentale pressione del tasto di Reset (se lo possedete), si recupera il vecchio programma. Attenzione: è necessario digitare i comandi su una sola riga SENZA farli precedere del numero di riga Basic.

```
POKE2050,1: SYS42291:
POKE45,PEEK(34):
POKE46,PEEK(35):CLR
```

Nota Bene

Alcune righe tra quelle pubblicate sembrano possedere più di 80 caratteri e, come tali, inaccettabili dal computer.

Nei casi in cui ci si accorga che la riga è troppo lunga, è necessario ricorrere alle abbreviazioni dei comandi così come indicato nell'appendice specifica riportata nel manuale del computer in vostro possesso.

Ad esempio invece di scrivere PRINT è possibile abbreviare col punto interrogativo (?). Invece di POKE potete scrivere il carattere "P" seguito dal carattere che viene visualizzato premendo contemporaneamente il tasto shift insieme con "O". Tutte le abbreviazioni possibili, lo ripetiamo, sono riportate in una delle appendici di qualsiasi manuale Commodore.

Nel caso sbagliate a digitare i microlistati che superano, in lunghezza, gli ottanta caratteri (SYNTAX ERROR o altri tipi di errore), è necessario, per sicurezza, ribatterli per intero e non apportare modifiche alla riga visualizzata con l'istruzione LIST.

La partecipazione dei lettori è gradita e compensata, in caso di pubblicazione, con materiale della Systems editoriale (libri, fascicoli arretrati, abbonamenti, eccetera).

Inviate le vostre 1 RIGA, purchè nella misura di almeno dieci per volta. Non ci è infatti possibile raggruppare i (moltissimi) micro-programmi che pervengono singolarmente in Redazione.

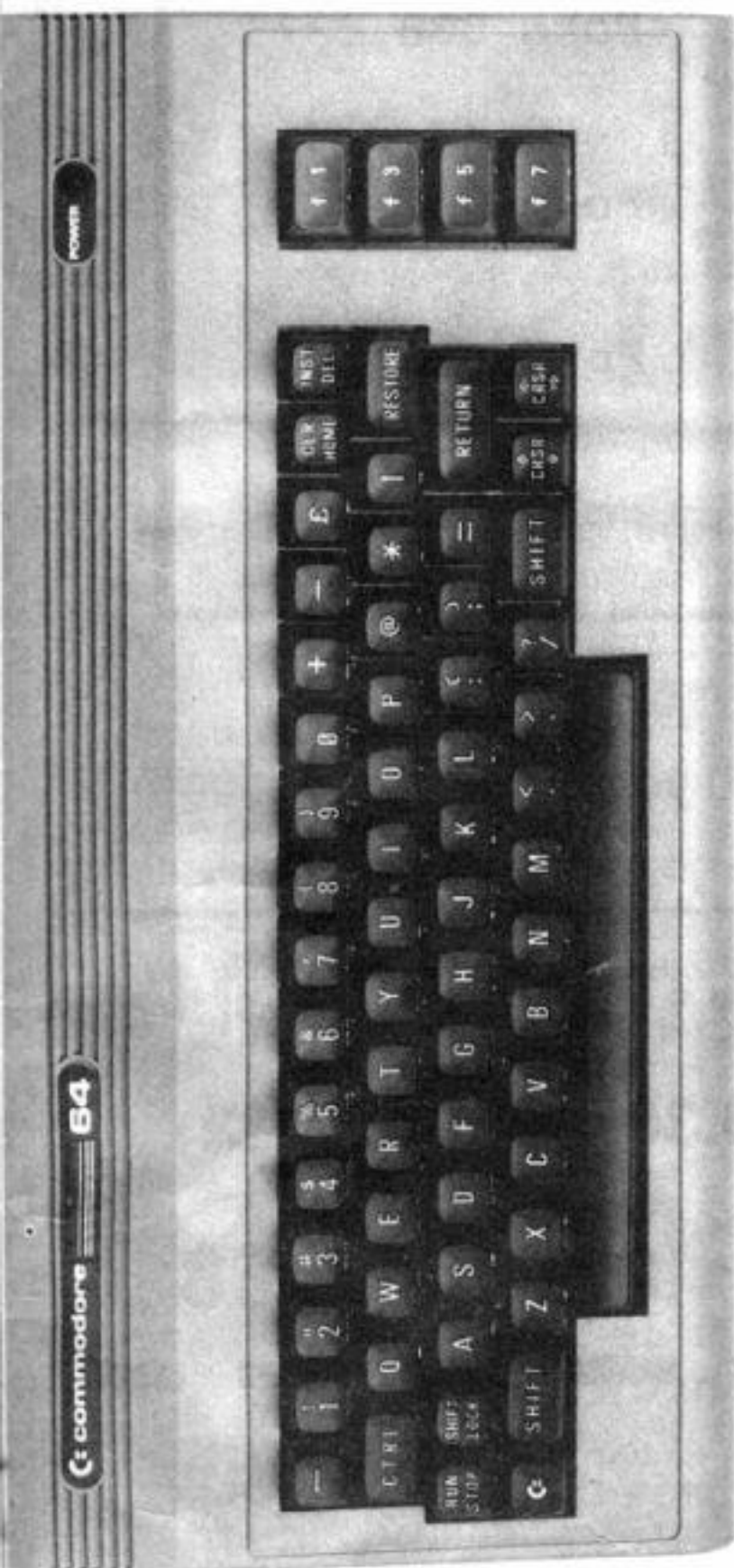
Inviate i vostri lavori su carta, (meglio se su nastro) corredati, ciascuno, di una breve spiegazione sulla funzione che compiono, proprio come li vedete pubblicati in queste pagine.

Ricordate di indicare chiaramente nome, cognome, via e città, e indirizzate a:

Commodore Computer Club
SYSTEMS EDITORIALE

Rubrica "1 Riga"

Viale Famagosta, 75
20142 MILANO

COME METTERE D'ACCORDO UN DO CON UN BIT.

No, non è fantascienza. Stiamo solo parlando delle fantastiche tastiere elettroniche Antonelli. Una novità che ti permette di dialogare con l'interfaccia di un computer per comporre, memorizzare e riascoltare tutta la musica che vuoi. E questo è solo l'inizio. Le tastiere elettroniche Antonelli sono una vera orchestra con tantissimi strumenti e voci diverse, possono essere collegate a impianti hi-fi per ottenere una fedeltà e una potenza di suono mai viste. In più sono facili da usare e belle da mostrare agli amici. Il prezzo? Questo sì è da fantascienza: da L. 250.000 (IVA esclusa).

Vuoi sapere di più sulle tastiere elettroniche Antonelli?

Compila e spedisci questo coupon a: ANTONELLI
Via Filottrano - 60027 Osimo (AN). Riceverai un completo catalogo.

NOME ______ COGNOME ______

VIA ______ N° ______

CAP. ______ CITTA' ______

Antonelli 2614 STEREO

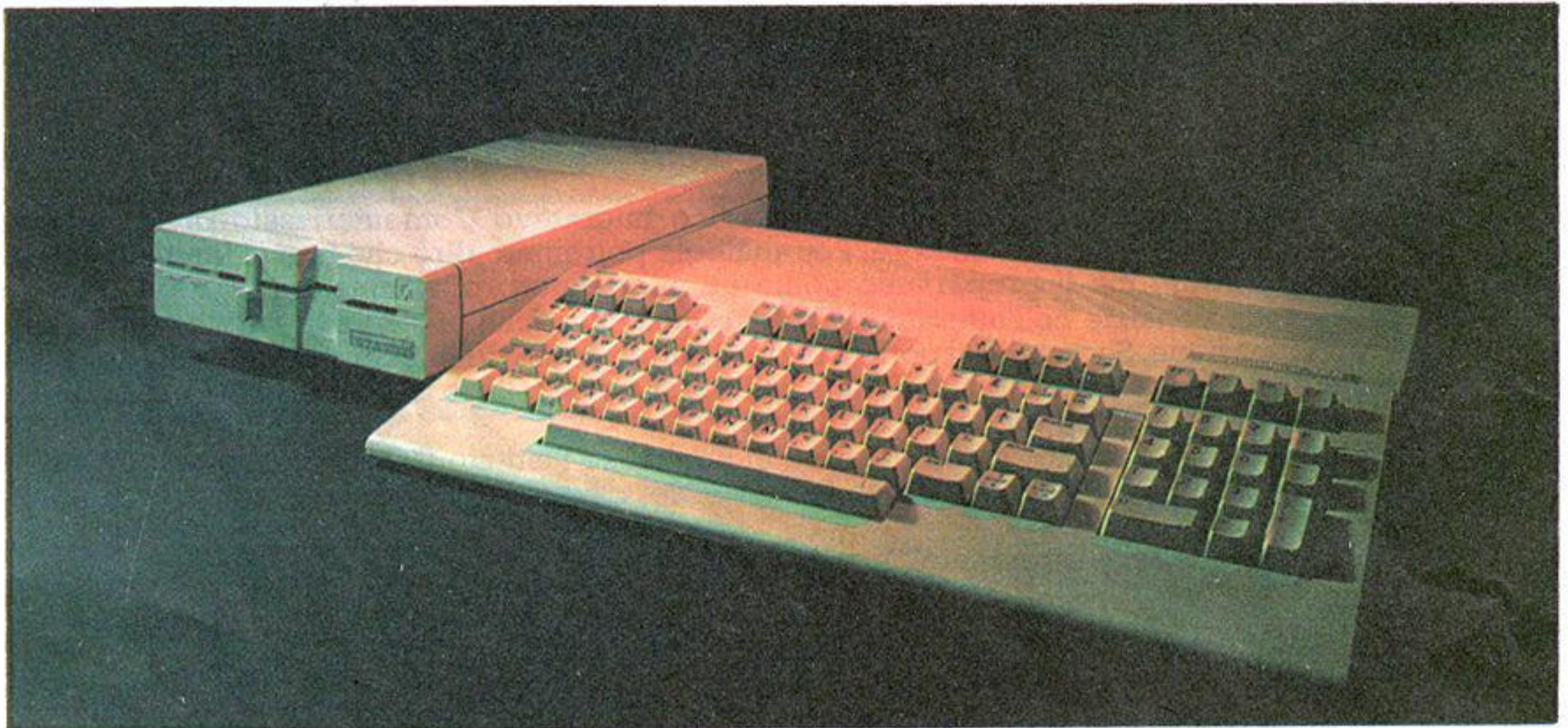
Uso di alcune Poke per C/64 di Giovanni Chiappe - Imperia

Ecco una simpatica tabella da sperimentare ed utilizzare in più occasioni:

Comando	Effetto	Ripristino
Poke 775,200	Inibisce comando List	Poke 775,167
Poke 808,239	Disabilita tasto Run/Stop	Poke 808,237
Poke 808,225	Disabil. List Run/Stop e Rest.	Poke 808,237
Poke 649,0	Disabilita la tastiera	Poke 649,10
Poke 818,32	Disab. comandi Save e Load	Poke 818,237

C128, impressioni a "caldo"

Un breve articolo utile a chi è intenzionato ad acquistare il nuovo modello della Commodore.

di Claudio Baiocchi

Le precedenti esperienze dell'autore sono relative al Vic 20 ed al Plus/4. Vengono illustrate le prime impressioni (favorevoli e sfavorevoli) sul nuovo computer appena acquistato.

La tastiera

Il C/128 ha un aspetto decisamente più "professionale" rispetto ai modelli inferiori. A ciò contribuiscono sia il tastierino numerico sia i numerosi tasti di colore grigio posti in alto, suddivisi in quattro blocchi di quattro ciascuno.. Quelli dell'ultimo blocco svolgono funzioni diverse a seconda se sono premuti da soli o insieme col tasto Shift.

Il gruppo più a destra sostituisce i tasti F1...F8 già presenti su tutti i modelli inferiori. La novità, rispetto al Vic 20 e al C-64, è costituita dal fatto che la loro programmabilità è immediata, grazie al comando KEY già esistente, del resto, su C-16 e Plus/4.

I quattro tasti del blocco successivo (procedendo da destra verso sinistra) sono un duplicato dei tasti di movimento del cursore e si rivelano comodi rispetto ai due soli tasti (da premere con o senza Shift) presenti nel C-64 e Vic 20. La loro posizione, però, non è delle più accessibili ma, si sa, è solo questione di abitudine. Questi tasti, così come gli altri otto che descriveremo, e quelli del tastierino numerico, sono disattivati quando si lavora nel "C-64 mode" (che d'ora in poi chiameremo, più sbrigativamente, "modo 64").

Segue un tasto "No Scroll": premendolo durante lo scrolling di un listato oppure l'output di dati sullo schermo, la visualizzazione si arresta. Premendolo ancora l'output riprende con un effetto analogo a quello che si ottiene premendo

contemporaneamente i tasti Control e "S": questa possibilità era già presente sul Plus/4. Volendo, invece, limitarsi a rallentare lo scrolling, basta tenere premuto il tasto Cbm, il primo basso a sinistra, proprio come avveniva su C-16 e Plus/4. Dovrebbe esser noto che sul Vic 20 e sul C-64 è il tasto Control che svolge quest'ultimo ruolo.

Il tasto successivo è, come quello di Shift-Lock, un tasto che, attivabile solo in modo 128, resta inserito dopo la pressione: se abbassato, determina l'output su video a 80 colonne ma non è utilizzabile con un normale televisore: occorre lavorare con un monitor.

Segue il tasto Line-Feed, utilizzabile solo in modo CP/M, mentre il quarto tasto del blocco è quello di Help: tale comando, già implementato nel Basic del C-16 e del Plus/4, è utilissimo in fase di debugging. Riservargli un tasto può sembrare eccessivo, dato che basta digitare "H" seguita da Shift + "E" e dal tasto Return per ottenere lo stesso effetto, ma su tale punto torneremo più avanti.

Segue il tasto Caps-Lock, simile allo Shift Lock dato che viene disinserito dalla successiva pressione, ma comunque non un doppione. Agisce, infatti, solo sulle lettere, e non sulla prima riga della tastiera (numeri e simboli sovrastanti). Segue il tasto Alt di cui il manuale dà una descrizione assolutamente inutilizzabile: apparentemente dovrebbe servire a "ridefinire" la tastiera ma, come avveniva per i tasti F1...F8 sul Vic 20 e C-64, non c'è il minimo accenno al suo uso.

Il penultimo tasto è Tab: premendolo, il cursore si sposta alla posizione della tabulazione successiva. Anche qui il manuale accenna alla possibilità di fissare autonomamente le tabulazioni, ma non dice come fare. I tasti Tab e Alt dovrebbero, comunque, esser utilizzati, più che altro, in modo CP/M.

L'ultimo tasto è l'Escape che non costituisce una novità per utenti di C-16 e Plus/4, salvo che per la posizione che occupa. Raccontare ai possessori di Vic 20 o C-64 tutte le interessantissime possibilità offerte da questo tasto è impossibile per mancanza di spazio.

Sul tastierino numerico c'è poco da dire: oltre alle cifre, contiene anche il punto (cioè la virgola) decimale, i tasti "+" e "—", ed un tasto Enter che funziona come il Return. Da notare che l'Enter ed il tasto dello zero sono di dimensioni doppie rispetto agli altri.

E con ciò abbiamo finito il giro dei tasti "nuovi".

Sui tasti "vecchi", oltre al fatto già segnalato, concernente il rallentamento dello scrolling (premere Cbm anzichè Control) c'è poco da dire: rispetto a C-16 e Plus/4 sono scomparsi i tasti di Flash-on e Flash-Off e, ovviamente, le comode funzioni associate. Altri tasti hanno ripreso la posizione che avevano su Vic 20 e C-64, come quelli di elevamento a potenza e di freccia a sinistra; è infine riapparso il tasto di Restore che, premuto insieme con Run/stop, è un'alternativa al pulsante di reset; alternativa talora più comoda, dato che il pulsante di reset, se premuto da solo, ha lo stesso effetto che spegnere e riaccendere il Computer; se è invece premuto insieme col tasto Run/stop non annulla il programma in memoria, ma porta in ambiente "Monitor" e... maltratta il puntatore allo stack: tornati al Basic (con X seguito da Return) è prudente fare un CLR.

Concludiamo la panoramica della tastiera proprio col tasto Run/Stop: come già sul C-16 e sul Plus/4, se viene premuto con lo Shift, provoca il caricamento e il successivo lancio (Run) del primo programma presente su disco anzichè su nastro, come per il Vic 20 ed il C-64. Naturalmente la variante è effettiva solo nel modo 128.

Si tratta, comunque, di una comodità pericolosa: se vogliamo, ad esempio, fermare un listing ed il dito che preme Run/stop tocca anche il tasto Shift, oppure lo Shift-Lock (che sono vicinissimi), si rischia di perdere il programma in memoria. Anche se si fa in tempo a premere ancora Run/Stop (da solo!) per evitare il caricamento del programma, non si può comunque evitare il successivo Run "incorporato" nel tasto, che farà quindi perdere tutti i dati elaborati prima dell'infortunio. La situazione è comunque leggermente migliorata rispetto a C-16 e Plus/4, dove anche il tasto Cbm, unito a Run/Stop, aveva lo stesso effetto. Si noti che precauzioni quali togliere il disco dal drive oppure tenerlo spento, pongono rimedio al primo guaio (non si perde il programma in memoria) ma non al secondo: eventuali dati già elaborati saranno comunque distrutti dal Run. Un trucco che permette di evitare tali rischi sarà indicato tra poco.

Il Basic

Terminato l'esame sommario della tastiera diamo un'occhiata, ancora più frettolosa, al Basic.

L'elenco delle parole riservate è impressionante: ben 171 elementi, anche se alcuni (Off e Quit) sono seguiti da un asterisco ad indicare che non sono in realtà implementati.

Altri comandi sono un... falso in atto pubblico, almeno se le cose funzionano come sui modelli precedenti: ad esempio Get # non è un vero comando (è codificato con i due token separati di Get e di #); idem dicasi di GetKey, che può essere abbreviato spezzando le abbreviazioni di Get e di Key. Ciononostante il panorama si presenta interessante: parecchi comandi relativi alla gestione del drive, riconoscibili dall'iniziale "D" (oltre a Dload, Dsave, DS, DS$, Directory già presenti su C-16 e Plus/4), troviamo Dclear, Dclose, Dopen, Dsave, Dverify. C'è anche un Catalog equivalente a Directory, ed inoltre Append, Collect, Concat, Copy, Header, Record, Rename, Scratch, che permettono di ottenere, con sintassi elementare, risultati comunque ottenibili sul Vic 20 e C-64, anche se con complicate operazioni attraverso il canale 15 del drive.

Vi sono anche comandi decisamente nuovi quali Bload, Boot, Bsave, e Run seguito dal nome di un programma.

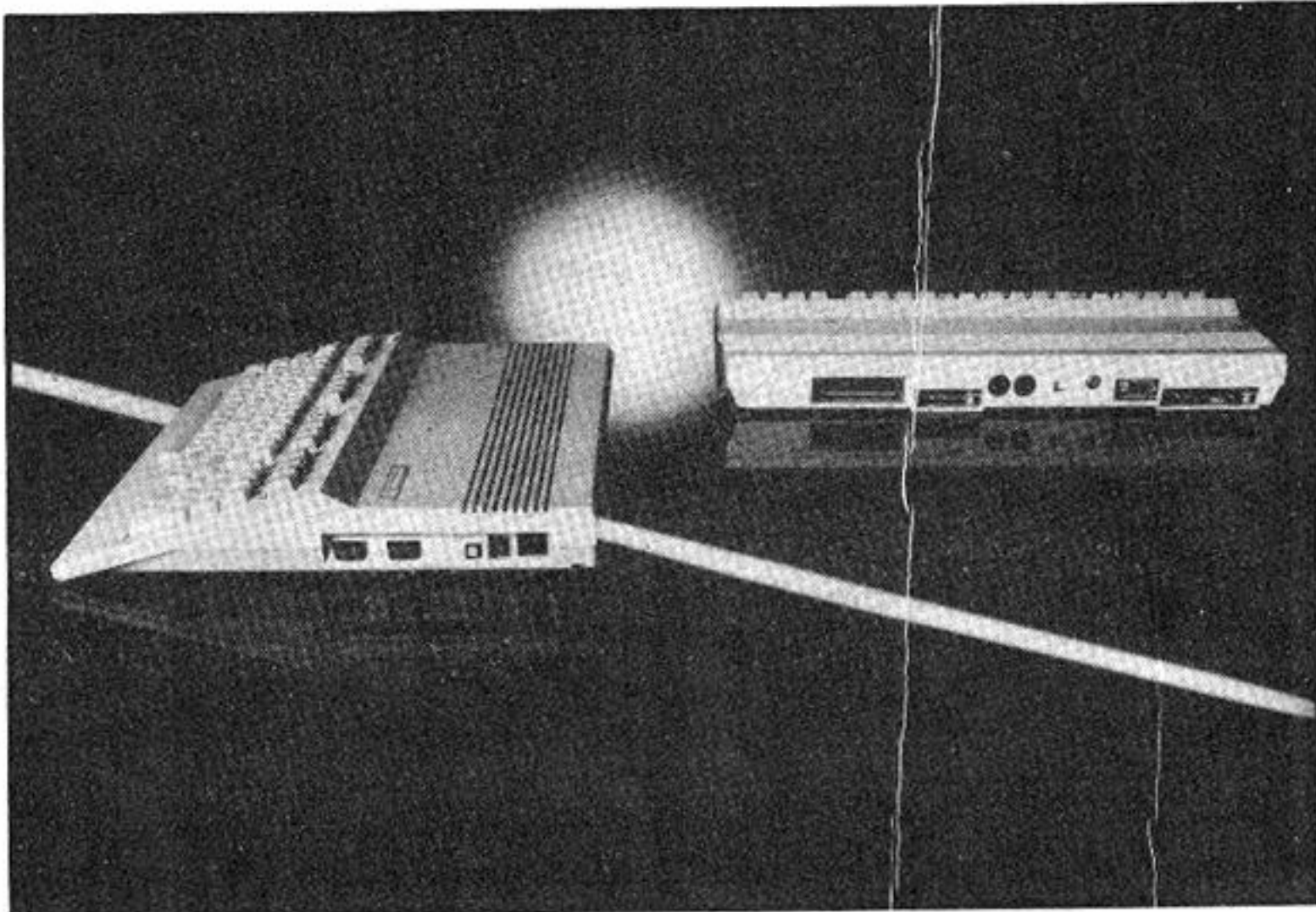
Molto ampliata, anche rispetto a C-16 e Plus/4 dove comunque era già vasta, è

la gamma dei comandi dedicati al suono e alla grafica. Pur non avendo avuto il tempo di sperimentare tutti i nuovi comandi, non ho saputo resistere alla tentazione di sperimentarne alcuni associati alla gestione degli sprite. Ho provato una certa delusione nel constatare che i comandi Bump e Collision sono scarsamente utilizzabili dato che non serve a nulla scoprire rapidamente che una collisione è avvenuta, se poi si prendono le decisioni conseguenti con la lentezza cronica del Basic. Entusiasmo suscita, invece, la facilità con cui si possono definire, modificare ed animare gli sprite; in particolare lo "Sprdef-Mode" è fantastico...

Da un punto di vista meno ludico, è apprezzabile la presenza di comandi strutturati come Begin...Bend all'interno di una IF o di una Else (oltre ai vari DO, Loop, While, Until già presenti su C-16 e Plus/4); comandi quali Fetch, Bank, Pointer, Stash, Swap (oltre ad un MONITOR più ampio di quello di C-16 e Plus/4) saranno certo molto apprezzati dagli "smanettoni" e il fatto che la descrizione del manuale sia molto scarna sarà per essi una sfida...

Interessante è infine la presenza del comando Fast (annullato da Slow) che, oscurando lo schermo e risparmiando di conseguenza il tempo dedicato alla sua gestione, porta ad un sensibile aumento della velocità di esecuzione. Da test effettuati su cicli For...Next sembra che la velocità doppia cui procede il clock corrisponda effettivamente ad una velocità quasi doppia di esecuzione del Basic. Tale risultato era ottenibile anche su C-16 e Plus/4, ma a prezzo di complicate manovre di Poke.

Notiamo ancora la possibilità di lettura diretta di joystick, paddle e penna luminosa (senza necessità di PEEK-are qua e là) e la presenza di comandi di aiuto alla stesura ed al debugging dei programmi, del resto già presenti su C-16 e Plus/4. Citiamo solo Auto, Delete, Renumber, Help, Tron, Troff, Trap, Resume.

Apprezzata è la presenza (finalmente!) dell'operatore XOR (cioè un Or esclusivo) e la possibilità di output formattati grazie a Using e Pudef.

Chiudiamo questa breve carrellata sul Basic con una avvertenza: purtroppo numerose abbreviazioni cui gli incalliti utenti Commodore erano abituati, sono da abbandonare: "P" seguita da Shift + E non abbrevia più PEEK, bensì PEN; P seguita da Shift + O sta per POT e non per POKE e così via. E' una scelta incomprensibile, che i progettisti della Commodore avrebbero potuto evitare con un minimo sforzo. Altro imbarazzo potrebbe creare la nuova sintassi della funzione FRE, associata alla ripartizione a priori della RAM in una zona-programma (FRE(0) informa sulla memoria residua per la scrittura di nuove linee Basic) e in una zona-variabili (FRE(1) fornisce lo spazio a disposizione per nuove variabili).

Questa rapida carrellata volge ormai al termine e la concludo con un trucco da PEEK-POKE-maniaco: per evitare i guai descritti, relativi al tasto Run/Stop, impariamo a disattivarlo eseguendo:

```
FOR X=4159 TO 4167:
POKE X,32:NEXT
```

In realtà si può fare di meglio: invece di disattivare il tasto, possiamo riprogrammarlo. In modo analogo possiamo riprogrammare addirittura il tasto HELP. In altre parole: gli 8 tasti funzionali sono in realtà 10....

Benchè l'operazione sia più complicata della programmazione dei tasti F1...F8 (per questi basta lavorare in Basic con il comando Key), possiamo in effetti, programmare anche i tasti Run/stop ed Help. Passando al MONITOR e digitando:

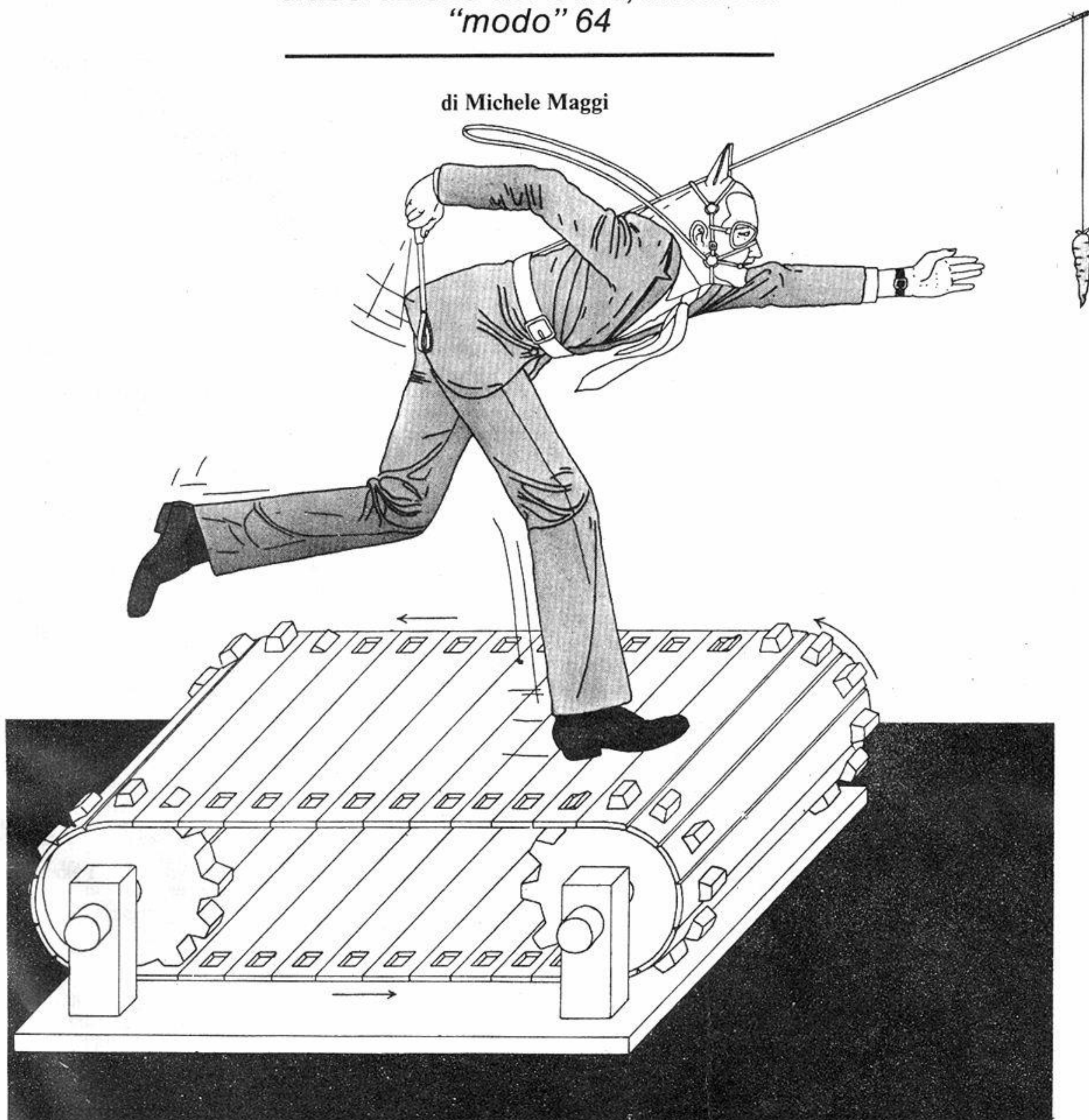
```
M 1000 (Return)
```

si vedranno apparire, a partire dalla locazione $100A, le stringhe associate a F1...F8, seguite da una stringa che è quella associata a Run/Stop che è l'abbreviazione di Dload, seguita da "*.RUN. Il puntino sta per Return. Subito dopo si trova la stringa HELP. Nelle locazioni da $1000 a $1009 sono riportate le lunghezze delle 10 stringhe.

L'operazione di disattivazione suggerita corrisponde a inserire un gruppo di "spazi" nella stringa associata a Run/Stop. Naturalmente, modificando (da MONITOR o da Basic) le lunghezze ed i contenuti di quelli che possiamo a buon diritto chiamare "tasti funzionali F9 e F10", possiamo fare in modo che un qualsiasi comando sia associabile ai dieci tasti funzione.

Un piccolo segreto del C128

Come aumentare la velocità di elaborazione del C128, anche in "modo" 64

di Michele Maggi

Come risulta dal manuale in dotazione del C128, esiste la possibilità di stabilire la frequenza di lavoro del computer mediante due istruzioni:

FAST che impone al calcolatore di lavorare alla frequenza di 2 megahertz (MHz).

Slow che lo riporta alla condizione normale, vale a dire quella di un solo MHz.

In quali casi conviene utilizzare l'una o l'altra frequenza di lavoro?

Notiamo, anzitutto, che quando un programma Basic incontra l'istruzione "Fast", accelera notevolmente l'esecuzione delle istruzioni successive, tranne quelle di Input/Output (I/O) che, vincolate ovviamente alla velocità intrinseca delle periferiche (stampante, drive), non possono venire accelerate in alcun modo.

Il principale inconveniente causato dall'istruzione "Fast" è che si perde completamente il controllo del video perché la gestione dello schermo è una delle operazioni che richiedono più tempo.

Questo fatto spiega perché, durante il caricamento di programmi da nastro, il video "scompare", e anche perché molte utility del tipo Turbodisk (equivalenti ai vari Turbotape, ma per floppy disk) fanno scomparire il video per accelerare le operazioni di caricamento.

L'istruzione Fast, quindi, si rivela utilissima quando all'interno di un programma, vengono eseguite elaborazioni molto lunghe e complesse che, inevitabilmente, costringono a notevoli attese.

Un esempio classico può essere l'ordinamento alfabetico o numerico di un vettore che, a parte il ricorso a varie tecniche che possano accelerarne l'esecuzione, rimane sicuramente una delle operazioni più lunghe in assoluto.

Grazie a "Fast" il tempo di ordinamento di un vettore viene praticamente dimezzato. Compiuta l'elaborazione si può riprendere il controllo del video con un semplice comando di "Slow".

FAST in modo C64

Tutto quanto è stato detto finora, però, non costituisce una novità dato che si riferisce ad informazioni riportate sul manuale. Il caso vuole, però, che esista la possibilità di portare il C128, operante in modo 64, a lavorare a 2 MHz come se gli fosse stato impartito il comando "Fast", non supportato, come è noto, dal misero Basic 2.0.

Vediamo ora come si può simulare il comando con una semplicissima POKE.

La locazione 53296 (in modo 64) contiene, normalmente, il valore 252. Inserendo, mediante Poke, il valore 253, perdiamo il video (nel senso che appariranno figure del tutto incomprensibili) ma avremo la possibilità di lavorare ad una velocità nettamente superiore al normale. È ovvio che, terminata l'elaborazione, sarà possibile riprendere il controllo del video con:
POKE 53296,252

Quanto affermato vale, però solo per il C64 "contenuto" nel C128 e non per un C64 "normale". Tale considerazione potrebbe far nascere una polemica sulla reale compatibilità tra C128, utilizzato in modo 64, e il software destinato al Commodore 64. C'è qualche lettore in grado di far notare altre differenze, più o meno vistose, in tal senso?

Il piccolissimo programma di sort (ordinamento di un vettore) ha lo scopo di dimostrare quanto detto finora.

Il programma

Si tratta di un semplice e banale Sort che non ha particolari pretese di velocità ed è anzi per questo che fa al caso nostro. Il suo funzionamento si basa sulla generazione di N numeri casuali ("N" è stabilito dall'utente) e sul loro successivo ordinamento in senso crescente.

Alla fine dell'elaborazione viene visualizzato il tempo impiegato e, subito dopo, vengono visualizzati i valori. In modo 128 "normale", cioè senza Fast, per ordinare 100 numeri si impiegano poco meno di 3 minuti, mentre col comando velocizzatore, occorre circa un minuto e mezzo.

Il programma si basa sull'ordinamento di numeri generati casualmente e il tempo impiegato ad ordinarli è influenzato notevolmente dal grado di "disordine degli stessi valori. In modo 64, senza l'istruzione POKE 53296,253, occorrono mediamente due minuti e mezzo mentre, se si aggiunge l'istruzione in questione, il tempo scende a un minuto e dieci secondi circa.

Anche per i tempi ottenuti in modo 64 valgono le considerazioni precedenti, pur se è possibile osservare che in modo 64 le operazioni si svolgono con una velocità lievemente superiore.

Funzionamento del programma

Le linee 10-30 si occupano della richiesta del numero dei valori da generare casualmente e da ordinare subito dopo. Viene anche effettuato un controllo sul numero digitato, e dimensionato il vettore N (E).

La riga 40 genera numeri casuali

La linea 50 inizializza la variabile TI$.

Le linee 60-100 eseguono il sort vero e proprio (che non illustriamo in dettaglio perché non è questo lo scopo del presente articolo).

Le linee 110-130 visualizzano il tempo TI$ e i valori ordinati.

Per velocizzare il tutto basterà aggiungere le seguenti linee:
In modo 128:
55 FAST e
115 SLOW
In modo 64:
55 POKE 53296,253 e
115 POKE 53296,252

Altre prove possono esser fatte dal lettore, tra cui suggeriamo quella di verificare il corretto funzionamento dell'orologio interno che sovraintende alla gestione delle variabili "TI" e "TI$.

```
  1 REM ORDINAMENTO IN SENSO CR
    ESCENTE DI UN VETTORE NUMER
    ICO
  2 :
 10 INPUT "QUANTI VALORI";E
 20 IF E<2 OR E>7500 THEN 10
 30 DIM N(E):PRINTCHR$(147)
 40 FOR I=1 TO E:N(I)=INT(RND(1
    )*1000)+1:NEXTI
 50 PRINT"STO ORDINANDO I NUMER
    I...":TI$="000000"
 60 C=0
 70 FOR I=1 TO E-1:IF N(I)<=N(I
    +1) THEN 90
 80 M=N(I+1):N(I+1)=N(I):N(I)=M
    :C=1
 90 NEXTI
100 IF C<>0 THEN  60
110 PRINT"TEMPO IMPIEGATO"TI$:P
    RINT"PREMI UN TASTO"
120 GET A$:IF A$="" THEN 120
130 FOR I=1 TO E:PRINTN(I):NEXT
```

Ordinamento ultraveloce di un vettore

Come ridurre a pochi secondi un'elaborazione che può anche richiedere ore di lavoro.

di Fabio Sorgato

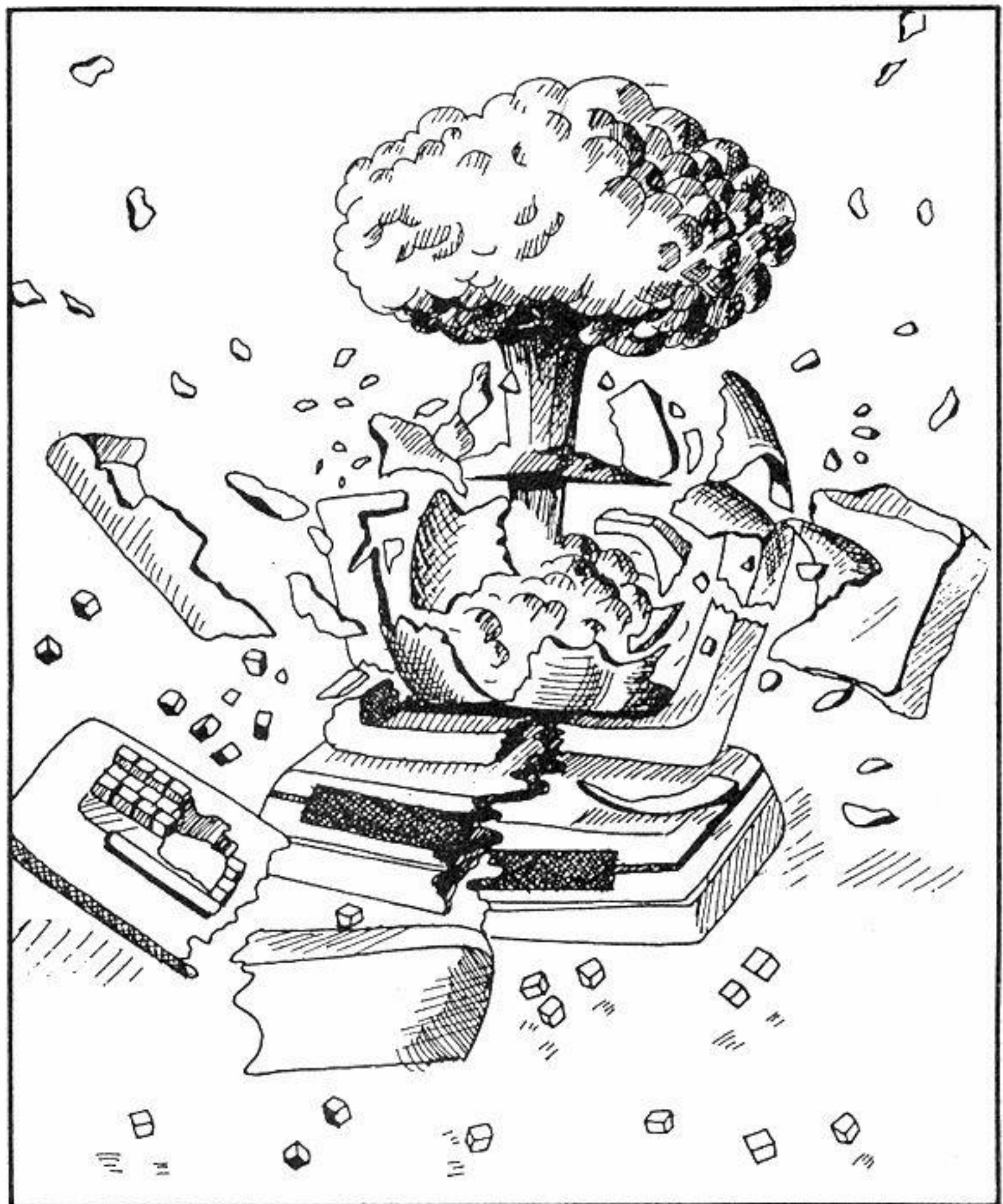
Il programma che presentiamo in queste pagine consente di ordinare in senso crescente tutti gli elementi di un vettore numerico predefinito. E' in grado, ovviamente, di sistemare anche gli elementi di un vettore stringa in ordine alfabetico.

Descrivere il funzionamento di questa routine in linguaggio macchina (LM) sarebbe impossibile. Ci limiteremo, pertanto, a decriverne l'utilizzo in modo tale che anche un principiante se ne possa servire.

- Caricate, anzitutto, il programma Sort pubblicato in queste pagine facendo bene attenzione a trascrivere con grande precisione tutti i valori riportati nei Data: un minimo errore può portare all'inchiodamento del sistema.

- Registrate e verificate il lavoro fatto. In caso di errore sarà più agevole ricaricarlo e rintracciare lo sbaglio.

- Dopo aver dato il Run, attendere alcuni secondi l'allocazione dei Data. Se tutto è in ordine dovrebbe comparire, alla fine, il messaggio "Sort attivato". In caso contrario un mesto messaggio avvertirà di un errore di digitazione.

- Se, quindi, avete ben digitato il listato, da questo momento, fino a che non spegnerete l'apparecchio, avrete a disposizione un nuovo comando Basic che ha, come sintassi, la freccia a sinistra (primo

tasto in alto a sinistra) seguito dal nome della variabile assegnata al vettore che si intende ordinare in senso crescente.

Per meglio comprendere il funzionamento del nuovo comando, caricate il programma Demo e studiatelo con attenzione:

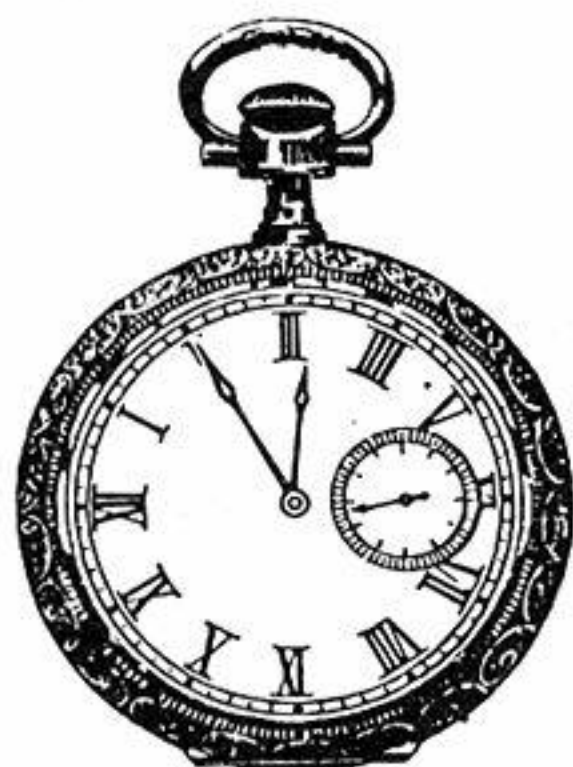
Riga 12: chiede il numero di elementi che desiderate assegnare al vettore stringa A$ (attenzione a non esagerare, per non incorrere in un "Out Of Memory Error").

Riga 20: crea gruppi casuali di stringhe alfanumeriche e le assegna ai singoli elementi del vettore.

Riga 50: attiva il nuovo comando e visualizza il tempo impiegato ad ordinare alfabeticamente (o numericamente) il vettore.

Riga 60: stampa su video il vettore perfettamente ordinato.

E' ovvio che il lettore, invece di A$, può indicare il nome che più gli aggrada.

L'unica limitazione è costituita dal fatto che non è possibile applicare il comando a vettori pluridimensionali.

ALLARME ROSSO

(per i principianti)

Nel digitare righe di programma basic che contengono istruzioni DATA, è piuttosto facile incorrere in errori di digitazione. Supponiamo che un'ipotetica linea basic numerata con 1200 debba contenere i tre valori: 123, 456, 789. Ecco alcuni esempi di errori più frequentemente commessi:

1200 DATA,123,456,789

C'è una virgola dopo la parola "DATA". I dati letti dal computer sono, in questo caso, quattro: 0, 123, 456, 789. Se, infatti, non figura alcun carattere dopo l'istruzione DATA, automaticamente viene assunto il valore nullo (0).

1200 DATA 123,456,789,

In questo caso, dopo il numero 789, il computer, grazie alla presenza della virgola erroneamente inserita, "crede" che ci sia un altro valore e, non trovandolo, lo assume come nullo (0).

1200 DATA 1234,56,789

La virgola è posizionata male, vale a dire dopo il carattere "4" e non dopo il carattere "3". Il computer non può sapere se il valore esatto è 123 oppure 1234 e individuare un errore, in questo caso, risulta piuttosto laborioso.

```
100 REM *********************
105 REM * TURBO SORT        *
110 REM * PER COMMODORE 64  *
115 REM * BY FABIO SORGATO  *
120 REM *********************
125 :
130 FOR K=0 TO 509:READ A
135 POKE 49152+K,A:REM SORT STRI
    NGHE
140 POKE 49664+K,A:REM SORT NUME
    RI
145 CK=CK+A:NEXT
150 IF CK=71855 THEN 160
155 PRINT"ERRORE DI DATI NEL 1'
    BLOCCO.":END
160 CK=0:FOR K=0 TO 100:READ A
165 POKE 49824+K,A:REM SOSTITUZI
    ONE DATI PER SORT NUMERI
170 CK=CK+A:NEXT
175 IF CK=14063 THEN 185
180 PRINT"ERRORE DI DATI NEL 2'
    BLOCCO.":END
185 CK=0:FOR K=0 TO 17:READ B,A
190 POKE 49512+B,A:REM SOSTITUZI
    ONE DATI PER SORT NUMERI
195 CK=CK+A+B:NEXT
200 IF CK=10262 THEN 210
205 PRINT"ERRORE DI DATI NEL 3'
```

```
    BLOCCO.":END
210 CK=0:FOR K=0 TO 132:READ A
215 POKE 50432+K,A:CK=CK+A:NEXT
220 IF CK=16610 THEN 230
225 PRINT"ERRORE DI DATI NEL 4'
    BLOCCO.":END
230 PRINTCHR$(147)"TURBO SORT AT
    TIVATO!":SYS50432:END
235 :
240 REM ********************
245 REM DATI PER ORDINAMENTO
250 REM ORDINAMNETO STRINGHE
255 REM ********************
260 :
265 DATA 165,047,133,251,165,048
270 DATA 133,252,165,050,197,252
275 DATA 208,006,165,049,197,251
280 DATA 240,047,160,000,177,251
285 DATA 201,065,208,007,200,177
290 DATA 251,201,193,240,037,169
295 DATA 000,133,092,160,002,177
300 DATA 251,024,101,251,144,002
305 DATA 230,092,170,200,177,251
310 DATA 024,101,252,024,101,092
315 DATA 133,252,134,251,076,008
320 DATA 192,162,027,076,139,227
325 DATA 160,002,169,000,133,092
330 DATA 177,251,024,101,251,144
335 DATA 002,230,092,133,096,200
340 DATA 177,251,024,101,252,024
345 DATA 101,092,133,097,165,251
350 DATA 024,105,007,144,002,230
355 DATA 252,133,251,164,097,165
360 DATA 096,056,233,003,176,001
365 DATA 136,133,096,132,097,032
370 DATA 165,193,169,000,133,255
375 DATA 165,251,133,098,165,252
380 DATA 133,099,165,098,133,251
385 DATA 165,099,133,252,165,096
390 DATA 133,253,165,097,133,254
395 DATA 169,128,133,002,160,000
400 DATA 177,251,240,072,133,088
405 DATA 200,177,251,133,089,200
410 DATA 177,251,133,090,160,000
415 DATA 177,253,240,077,133,091
420 DATA 200,177,253,133,092,200
425 DATA 177,253,133,093,169,000
430 DATA 133,095,165,088,197,091
435 DATA 176,007,133,094,230,095
440 DATA 024,144,004,165,091,133
445 DATA 094,160,000,177,089,209
450 DATA 092,208,011,200,196,094
455 DATA 208,245,165,095,240,004
460 DATA 208,025,144,023,160,000
465 DATA 177,251,170,177,253,145
470 DATA 251,138,145,253,200,192
475 DATA 003,208,241,165,002,073
480 DATA 128,133,002,165,002,208
485 DATA 014,165,251,024,105,003
490 DATA 144,002,230,252,133,251
495 DATA 024,144,011,165,253,056
500 DATA 233,003,176,002,198,254
505 DATA 133,253,165,252,197,254
510 DATA 176,003,076,160,192,208
515 DATA 006,165,251,197,253,144
520 DATA 245,165,252,197,097,144
525 DATA 010,240,002,176,037,165
530 DATA 251,197,096,176,031,230
535 DATA 255,166,255,164,252,165
540 DATA 251,024,105,003,144,001
545 DATA 200,157,000,207,152,157
550 DATA 064,207,165,096,157,128
555 DATA 207,165,097,157,192,207
560 DATA 164,252,165,251,056,233
565 DATA 003,176,001,136,133,096
570 DATA 132,097,165,099,197,097
575 DATA 176,003,076,140,192,208
580 DATA 006,165,098,197,096,144
585 DATA 245,165,255,240,027,166
590 DATA 255,189,000,207,133,098
595 DATA 189,064,207,133,099,189
```

```
600 DATA 128,207,133,096,189,192
605 DATA 207,133,097,198,255,076
610 DATA 140,192,096,030,060,090
615 DATA 120,165,251,133,088,165
620 DATA 252,133,089,173,117,220
625 DATA 041,003,170,165,089,133
630 DATA 254,189,161,193,024,101
635 DATA 088,144,002,230,254,133
640 DATA 253,165,254,197,097,240
645 DATA 004,144,008,176,023,165
650 DATA 253,197,096,176,017,160
655 DATA 000,177,088,170,177,253
660 DATA 145,088,138,145,253,200
665 DATA 192,003,208,241,165,088
670 DATA 024,105,003,144,002,230
675 DATA 089,133,088,165,089,197
680 DATA 097,144,184,165,088,197
685 DATA 096,144,178,096,000,000
690 :
695 REM ********************
700 REM DATI PER SOSTITUZIONE
705 REM PER ORDINAMENTO
710 REM NUMERI
715 REM ********************
720 :
725 DATA 160,001,177,253,041,128
730 DATA 133,091,177,251,041,128
735 DATA 208,006,165,091,240,017
740 DATA 208,024,165,091,240,077
745 DATA 032,230,194,165,093,201
750 DATA 002,208,068,240,009,032
755 DATA 230,194,165,093,201,001
760 DATA 208,057,160,000,177,251
765 DATA 170,177,253,145,251,138
770 DATA 145,253,200,192,005,208
775 DATA 241,165,002,073,128,133
780 DATA 002,024,144,026,160,000
785 DATA 132,093,177,251,209,253
790 DATA 208,006,200,192,005,208
795 DATA 245,096,144,003,230,093
800 DATA 096,169,002,133,093,096
805 DATA 234,234,234,234,234
810 :
815 DATA 279,195,269,005,218,194
820 DATA 184,065,640,005,633,005
825 DATA 593,195,572,200,571,150
830 DATA 570,100,569,050,567,194
835 DATA 528,194,512,005,485,005
840 DATA 450,194,435,005,421,005
845 :
850 REM *****************
855 REM DATI
860 REM PER RICONOSCIMENTO
865 REM NUOVO  COMANDO
870 REM *****************
875 :
880 DATA 169,011,141,008,003,169
885 DATA 197,141,009,003,096,032
890 DATA 115,000,201,095,240,006
895 DATA 032,121,000,076,231,167
900 DATA 169,000,133,252,165,122
905 DATA 133,254,165,123,133,255
910 DATA 160,001,162,004,177,254
915 DATA 201,036,240,024,201,065
920 DATA 144,020,201,091,176,011
925 DATA 201,058,240,012,153,250
930 DATA 000,200,202,208,231,162
935 DATA 011,076,139,227,170,152
940 DATA 024,101,122,144,002,230
945 DATA 123,133,122,160,001,224
950 DATA 036,208,027,185,250,000
955 DATA 141,025,192,200,185,250
960 DATA 000,009,128,141,032,192
965 DATA 032,000,192,230,122,208
970 DATA 002,230,123,076,174,167
975 DATA 185,250,000,141,025,194
980 DATA 200,185,250,000,141,032
985 DATA 194,032,000,194,076,174
990 DATA 167
```

```
10 REM  DEMO SORT
12 INPUT "VETTORE DI ELEMENTI";
   N
15 DIM A$(N):FOR K=0 TO N
20 R=RND(1)*5:A$="":FOR J=1 TO
   R:A$=A$+CHR$(RND(1)*26+65):N
   EXT
30 PRINTK;A$:A$(K)=A$:NEXT
40 PRINT:PRINT"PREMI UN TASTO":
   POKE 198,0:WAIT 198,1
50 TI$="000000":←A$:T=TI:PRINTT
   /60"SEC.":POKE 198,0:WAIT 19
   8,1
60 FOR K=0 TO N:PRINTA$(K):NEXT
```

Scassamuro

Un semplice giochino per acquistare familiarità con la tastiera.

di Maurizio Dell'Abate

Per lavorare in modo produttivo ed efficace col computer è necessaria, tra le altre cose, una buona, se non perfetta, padronanza della tastiera.

Questo è uno dei problemi più ardui da risolvere per i principianti, che devono "addestrarsi" notevolmente per colmare la lacuna. Farlo con un programma tipo "corso di dattilografia" o altre cose del genere può risultare noioso e stressante, mentre imparare per mezzo di un giochino lo è sicuramente di meno. E' proprio questo lo scopo di SCASSAMURO, un game che riesce ad essere anche utile.

Il listato gira sul C—64, sul C—16 e sul Plus/4 e non dovrebbero esistere problemi per la trascrizione in quanto è relativamente breve. Ricordiamo che le righe conenenti soltanto istruzioni REM possono non essere copiate: hanno soltanto funzione di chiarimento e non influiscono sul funzionamento del programma.

Per facilitare la lettura del listato sono stati eliminati tutti i simboli "strani" in reverse, sostituendoli con la corrispondente funzione CHR$.

Prima di dare il RUN, registrate e verificate il programma: un errore di copiatura (specialmente per le funzioni di POKE) può essere letale.

Come si gioca

Scopo del gioco è impedire che le 200 bombe che vengono sganciate distruggano completamente il muro posto in basso. Ciascuna bomba è, in realtà, una delle 26 lettere dell'alfabeto che, ad ogni buon conto, è riportato in basso sotto forma di tastiera ASCII. Per impedire che distruggano un pezzo di muro è necessario premere la lettera corrispondente prima dell'impatto.

All'inizio del gioco viene chiesto il livello di difficoltà che può essere 1, 2, 3, 4 oppure 5: 1 è il più difficile mentre il 5, ovviamente, è piuttosto semplice.
Il livello di difficoltà selezionato influisce per tre motivi:

- Varia la velocità della bomba
- Cambia la distanza del muro dal punto da cui partono le bombe;
- Muta lo spessore del muro: a parità di bombe sganciate è più facile che venga distrutto un muro sottile, che uno spesso.

Il gioco termina quando il muro viene distrutto completamente o quando vengono sganciate le 200 bombe.

Nel primo caso il programma vi comunica che non avete sufficiente dimestichezza con la tastiera; se invece le 200 bombe non sono bastate a decimare il muro, viene comunicato il punteggio. Questo è inversamente proporzionale al numero delle bombe che non sono state bloccate.

Possibili sofisticazioni

Il programma si presta molto bene per essere ampliato. Qualche idea:

- Inserire una routine sonora nell'ultima riga.
- Variare il numero delle bombe a seconda del livello di difficoltà.
- Fare in modo che un simbolo grafico abbia una funzione di jolly e possa quindi essere fermato premendo un tasto qualsiasi.
- Fare in modo che una lettera possa cambiare durante la caduta.

```
100 REM  SPACCAMURO
110 REM  GIOCO PER C-64 E C-16
120 :
130 REM  BY MAURIZIO DELL'ABATE
140 :
150 PRINTCHR$(147)"1 C-64":PRIN
    T"2 C-16 PLUS/4"
160 GET A$:IF A$="" THEN 160
170 IF A$="1" THEN SC=1024:GOTO
     220
180 IF A$="2" THEN SC=3072:GOTO
     220
190 GOTO 160
200 :
210 REM  SCHERMATA INIZIALE
220 PRINTCHR$(147);:PRINT:PRINT
230 PRINT" S C A S S A M U R O"
    ;CHR$(144)
240 FOR I=0 TO 39:PRINTCHR$(195
    );:NEXT
250 PRINT:PRINT" LIVELLO DI DIF
    FICOLTA'?"
260 PRINT:PRINT" 1=DIFFICILE ..
    ...... 5=FACILE"
270 GET A$:IF A$<"1" OR A$>"5"
    THEN 270
280 W=10*VAL(A$)-10:GOSUB 710:L
    V=VAL(A$):REM  W=RITARDO
290 PRINT:PRINTCHR$(144);" HAI
    SCELTO: ";LV
300 PRINT" PREMI UN TASTO PER I
    NIZIARE"
310 GET R$:IF LEN(R$)=0 THEN 31
    0
320 GOSUB 710
330 REM  INIZIO GIOCO
340 PRINTCHR$(147):PRINT
350 FOR I=0 TO 39:PRINTCHR$(195
    );:NEXT
360 FOR I=0 TO 8+LV:PRINT:NEXT
370 PRINTCHR$(144);CHR$(18);
380 FOR I=0 TO LV
390 FOR G=0 TO 39:PRINTCHR$(32)
    ;:NEXT
400 NEXT:PRINT:GOSUB 680
410 FOR I=0 TO 199:GOSUB 710
420 ZX=40*RND(1):AS=0
430 LX=SC+160+ZX:IF O<15 AND PE
    EK(LX+40*(8+(2*LV)))=32 THE
    N O=O+1:GOTO 420
440 O=0:CL=26*RND(1)+1
450 GET R$:FOR Y=0 TO 19:PRINTC
    HR$(19):PRINT" SGANCIATE";I
    +1;CHR$(157);" BOMBE"
460 IF CO=(LV+1)*40 THEN 610
470 FOR B=0 TO LV-1:GET A$:IF L
    EN(A$)>0 THEN AS=ASC(A$)
480 NEXT
490 IF AS=INT(CL)+64 THEN Y=19:
    GOSUB 710:GOTO 520
500 DF=LX+40*Y:IF PEEK(DF)=160
    THEN CO=CO+1:Y=19:POKE DF-4
    0,32:GOTO 520
510 POKE DF,CL:POKE DF-40,32:FO
    R HH=1 TO W:NEXT:REM  W=RIT
    ARDO
520 NEXT:POKE DF,32
530 NEXT
540 REM  MESSAGGI E PUNTEGGIO
550 PRINTCHR$(147);:PRINT:PRINT
    :PRINT
560 PRINT" OK, VA BENE."
570 PRINT:PRINT" IL TUO PUNTEGG
    IO E'";CHR$(144);INT((1/CO)
    *10000)
580 PRINT:PRINT" PREMI UN TASTO
     SE VUOI MIGLIORARE."
590 GET R$:IF LEN(R$)=0 THEN 59
    0
600 GOSUB 710: RUN
610 PRINTCHR$(147):PRINT:PRINT:
    PRINT
620 PRINT" I TUOI RAPPORTI CON
    LA TASTIERA"
630 PRINT" NON SONO TRA I MIGLI
    ORI"
640 PRINT:PRINT:PRINT" SEI RIUS
    CITO A FARTI ABBATTERE"
650 PRINT" IL MURO."
660 PRINT:PRINTCHR$(144)" CONTI
    NUA AD ALLENARTI!";:GOTO 57
    0
670 :
680 PRINT"Q W E R T Y U I O P"
690 PRINT" A S D F G H J K L"
700 PRINT"  Z X C V B N M";:RET
    URN
710 RETURN:REM  INSERIRE LA ROU
    TINE SONORA
```

C/64, C/16, Plus/4

Magic square

Un famoso rompicapo trasformato in video game, per la gioia dei Commodoriani e per il dolore del cervello.

di Maurizio Dell'Abate

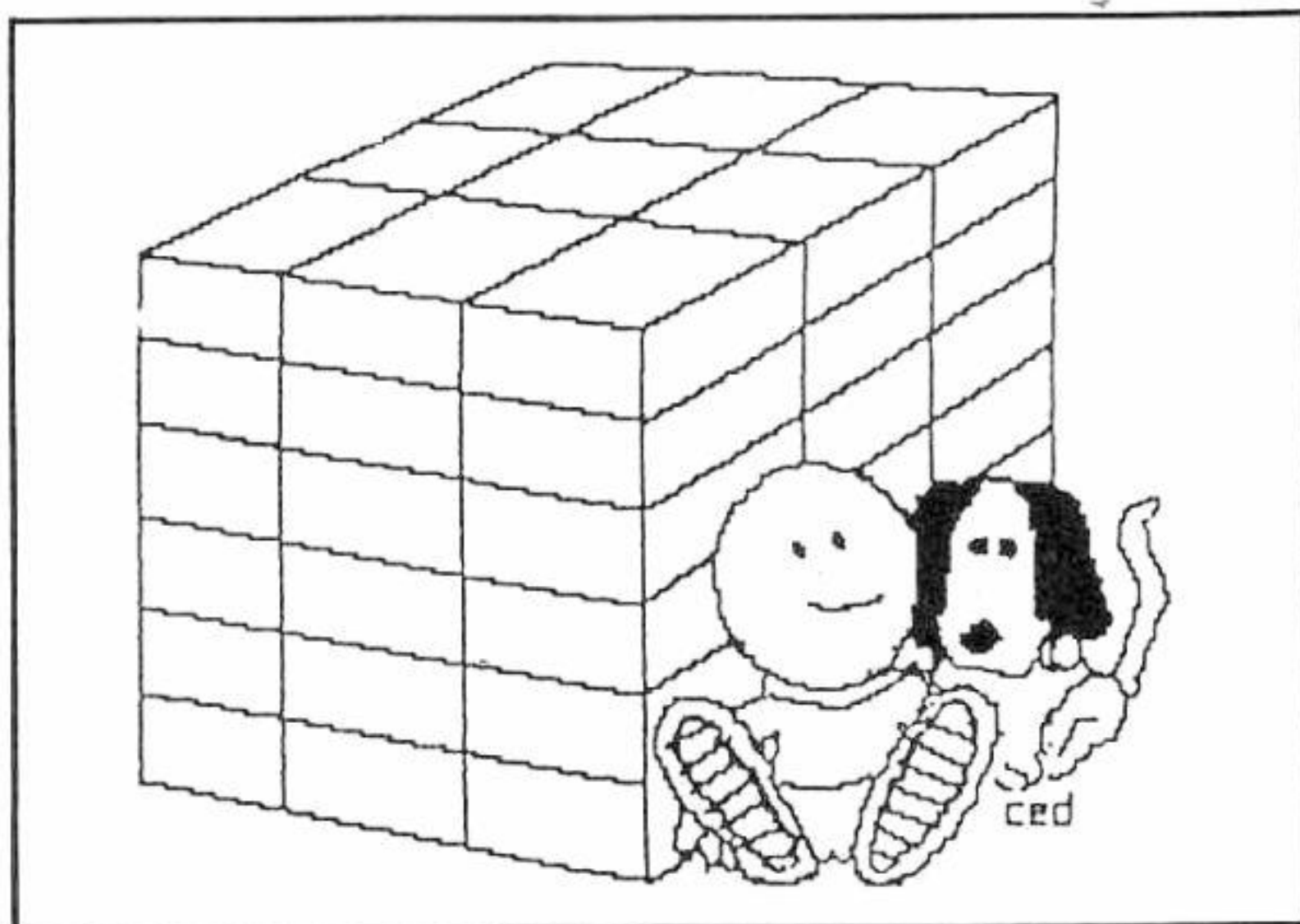
Questo gioco mescola in modo casuale le lettere dell'alfabeto dalla "A" alla "O" e le dispone in una tabella quadrata di quattro righe per lato per un totale di 16 caselle.

Una delle celle rimane vuota e scopo del gioco è quello di porre in ordine alfabetico le lettere facendole scorrere nel quadrato, analogamente al classico giochino con i numeri da 1 a 15.

Per muovere una lettera si deve premere, è ovvio, il tasto ad essa corrispondente. Se risulta "bloccata" a causa della lontananza dalla casella vuota, sul video compare il messaggio: "Movimento non possibile". Lo spazio vuoto deve, al termine del gioco, trovarsi nella cella più in basso a destra della griglia. In caso contrario il programma non sarà in grado di riconoscere la vostra fatica mentale.

Nel gioco le mosse vengono conteggiate e visualizzate, ma non potete effettuarne più di 300 (sembrano molte ma non lo sono affatto...). Superato questo limite il gioco termina irrisoluto.

E' stata introdotta un'altra sofisticazione, assente nel gioco "meccanico" originale. E' possibile invertire la posizione di una lettera con quella che si trova nella casella sottostante. Per effettuare questa mossa è sufficiente premere il tasto "S", seguito dalla lettera desiderata. Come è facile intuire, non è possibile operare uno scambio indicando una delle lettere presenti nell'ultima fila in basso. Un messaggio, in questo caso, segnalerà l'errore.

L'uso dell'inversione facilita però notevolmente il gioco, ed è quindi giusto

penalizzare il giocatore conteggiando ben quaranta mosse per ogni scambio effettuato. In questo modo non potrete ragionevolmente sfruttare la funzione di scambio troppo spesso nel corso della stessa partita.

La parte più difficile del gioco si verifica quando le ultime tre lettere (M, N, O) sono le uniche che devono essere ancora ordinate. Solo se utilizzerete con intelligenza l'inversione riuscirete a portare a termine il gioco. Ricordate che l'inversione agisce solo in senso verticale e, quindi, se vorrete invertire due lettere affiancate tra loro dovrete opportunamente spostarle.

Il gioco gira sui Commofore a 40 colonne, a patto però che segnaliate al programma quale computer state usando. I commenti al listato (REM), che in questo caso non sono pochi, possono tranquillamente essere tralasciati: le linee di commento non vengono infatti "chiamate" da istruzioni del tipo GOTO oppure GOSUB.

Il programma non prevede suoni, colori, o particolari effetti grafici allo scopo di non compromettere la compatibilità: è compito vostro, se lo desiderate, rendere più gradevole il programma.

Concludiamo ricordando la corretta disposizione delle lettere affinchè il rompicapo sia considerato completo:

A B C D
E F G H
I J K L
M N O

```
100 REM QUADRATO MAGICO
105 :
110 REM BY MAURIZIO DELL'ABATE.
115 REM PER C64, C16, PLUS/4
120 :
125 PRINTCHR$(147):PRINT" 1. C6
    4 E C128 MODO 64"
130 PRINT:PRINT" 2. C16 E PLUS/
    4":PRINT:PRINT" SCEGLI..."
135 GET O$:IF O$<"1" OR O$>"2"
    THEN 135
140 IF O$="2" AND PEEK(1)=55 TH
    EN E=1:PRINT"SECONDO ME HAI
     IL 64... VEDREMO!!"
145 IF E=1 THEN FOR RE=0 TO 300
    0:NEXT
150 LM=300:REM NUMERO-LIMITE DI
     MOSSE
155 EN=1000:DIM L$(3,3):PRINTCH
    R$(147)
160 PRINT:PRINT:PRINT TAB(11);"
    ______________":REM 14 VOLT
    E TASTO COMMODORE + @
165 PRINT TAB(11)CHR$(18)" MAGI
    C SQUARE "
170 PRINT:PRINT TAB(8)"OVVERO:
    GIOCO DEL 15 ":PRINT
175 FOR I=0 TO 39:PRINT"=";:NEX
    T:PRINT:PRINT
180 REM ***************
185 REM USA SHIFT + C, SHIFT +
    -
190 REM COMMODORE + A, S, Z, X
195 REM COMMODORE + Q, W, E, R
200 REM SHIFT + +
205 REM ***************
210 PRINT TAB(14);"┌─┬─┬─┬─┐"
215 FOR I=0 TO 2
220 PRINT TAB(14);"│ │ │ │ │"
225 PRINT TAB(14);"├─┼─┼─┼─┤"
230 NEXT
235 PRINT TAB(14);"│ │ │ │ │"
240 PRINT TAB(14);"└─┴─┴─┴─┘":P
    RINT
245 FOR I=0 TO 39:PRINT"=";:NEX
    T:PRINT:PRINT
250 :
255 REM  INIZIO CASUALE
260 :
265 FOR I=65 TO 79
270 NC=4*RND(1):ND=4*RND(1):IF
    L$(NC,ND)<>"" THEN 270
275 L$(NC,ND)=CHR$(I):NEXT
280 :
285 REM  INSERISCE LO SPAZIO
290 :
295 FOR B=0 TO 3
300 FOR I=0 TO 3
305 IF L$(I,B)="" THEN L$(I,B)=
    CHR$(32):GOTO 330
310 NEXT:NEXT
315 :
320 REM STAMPA E CONTROLLA
325 :
330 V=0:CC=0:X=15:Y=12:FOR B=0
    TO 3:FOR I=0 TO 3
335 A$=L$(I,B):GOSUB 665:IF A$=
    CHR$(65+CC) THEN V=V+1
340 CC=CC+1:X=X+2:IF X=23 THEN
    X=15:Y=Y+2
345 NEXT:NEXT
350 IF V=15 THEN 610:REM  VITTO
    RIA
355 IF MS>=LM THEN 635:REM  SCO
    NFITTA
360 :
365 REM LEGGE LA TASTIERA
370 :
375 GET A$
380 IF A$=CHR$(83) THEN 510:REM
      CAMBIO
385 IF A$<"A" OR A$>"O" THEN 37
    5
390 :
395 REM E' POSSIBILE LO SPOSTAM
    ENTO?
400 :
405 FOR B=0 TO 3:FOR I=0 TO 3:I
    F L$(I,B)<>A$ THEN NEXT:NEX
    T
410 Z=I-1:IF Z<0 THEN 420
415 IF L$(Z,B)=CHR$(32) THEN 47
    0:REM  POSSIBILE MUOVERSI A
     SINISTRA
420 Z=I+1:IF Z>3 THEN 430
425 IF L$(Z,B)=CHR$(32) THEN 47
    0:REM  POSSIBILE MUOVERSI A
     DESTRA
```

```
430 X=B-1:IF X<0 THEN 440
435 IF L$(I,X)=CHR$(32) THEN 49
    0:REM  POSSIBILE MUOVERSI I
    N ALTO
440 X=B+1:IF X>3 THEN 450
445 IF L$(I,X)=CHR$(32) THEN 49
    0:REM  POSSIBILE MUOVERSI I
    N BASSO
450 A$="MOVIMENTO NON POSSIBILE
               ":X=1:Y=1:GOSUB
     665:GOTO 375
455 :
460 REM A SINISTRA E A DESTRA
465 :
470 L$(I,B)=CHR$(32):L$(Z,B)=A$
    :GOTO 580
475 :
480 REM IN ALTO E IN BASSO
485 :
490 L$(I,B)=CHR$(32):L$(I,X)=A$
    :GOTO 580
495 :
500 REM SCAMBIO DELLE LETTERE
505 :
510 A$="SCAMBIO TRA LA CASELLA
    RICHIESTA...":X=1:Y=1:GOSUB
     665:FOR Q=1 TO EN:NEXT
515 A$="E LA CASELLA SOTTOSTANT
    E.          ":X=1:Y=1:GOSUB
     665:FOR Q=1 TO EN:NEXT
520 A$="QUALE CASELLA DESIDERI?
     (PREMI)     ":X=1:Y=1:GOSUB
     665:FOR Q=1 TO EN:NEXT
525 GET A$
530 IF A$=CHR$(32) THEN 540
535 IF A$<"A" OR A$>"O" THEN 52
    5
540 FOR B=0 TO 3:FOR I=0 TO 3:I
    F L$(I,B)<>A$ THEN NEXT:NEX
    T
545 NB=B+1:IF NB>3 THEN 450
550 C$=L$(I,B):L$(I,B)=L$(I,NB)
    :L$(I,NB)=C$
555 MS=MS+40:A$="TI PENALIZZO C
    ON40 MOSSE. MOSSE:"+STR$(MS
    ):X=1:Y=1:GOSUB 665
560 GOTO 330
565 :
570 REM AGGIORNA E STAMPA LE MO
    SSE
575 :
580 JH$="        ":REM  8 SPAZI
     BIANCHI
585 MS=MS+1:A$="OK. MOSSA VALID
    A. MOSSE:"+STR$(MS)+JH$:X=1
    :Y=1:GOSUB 665
590 GOTO 330
595 :
600 REM CASO DI VITTORIA...
605 :
610 A$=CHR$(18)+"HAI VINTO!!!!!
    !!":X=1:Y=1:GOSUB 665:FOR I
    =1 TO 2000:NEXT
615 A$="CE L'HAI FATTA IN"+STR$
    (MS)+" MOSSE.    ":X=1:Y=1:G
    OSUB 665:GOTO 695
620 :
625 REM ...E DI SCONFITTA
630 :
635 FOR I=1 TO 2000:NEXT
640 A$=CHR$(18)+"HAI PERSO!!!!!
    !!":X=1:Y=1:GOSUB 665:FOR I
    =1 TO 2000:NEXT
645 A$="TROPPE MOSSE. ALLENATI
    E MIGLIORERAI!":X=1:Y=1:GOS
    UB 665:GOTO 695
650 :
655 REM POSIZIONA IL CURSORE A
    X,Y
660 :
665 IF O$="2" THEN 675
670 POKE 783,PEEK(783) AND 254:
    POKE 781,Y:POKE 782,X:SYS65
    520:PRINTA$:RETURN
675 CHAR,X,Y-1,A$:RETURN
680 :
685 REM PARTE CONCLUSIVA
690 :
695 FOR I=1 TO 2000:NEXT
700 A$="PREMI * PER GIOCARE DI
    NUOVO":X=1:Y=1:GOSUB 665
705 GET A$:IF A$<>CHR$(42) THEN
     705
710  RUN
715 END
```

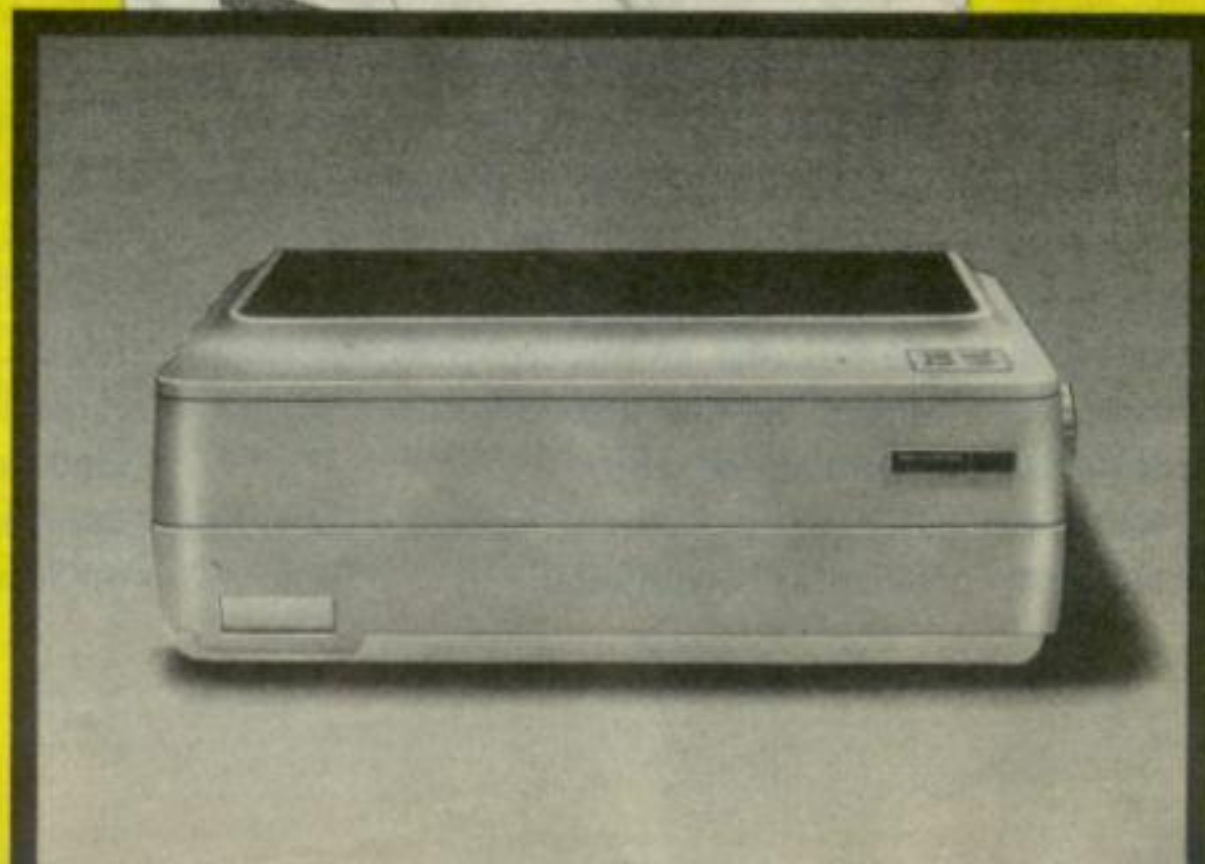
O.K. NON HO DUBBI. AL MIO PC COLLEGO LA MT/86

Le stampanti **MT/86, a 136 colonne e MT/85, a 80 colonne**, rappresentano una nuova frontiera nel settore delle stampanti a basso costo.

Basso costo, ma non bassa qualità e basse prestazioni, infatti ecco le credenziali di questi due nuovi prodotti.

Velocità a 180 cps. bidirezionale ottimizzata, NLQ a 45 cps., grafiche, possibilità 8 fonti alternative di caratteri e naturalmente **la completa compatibilità con i ... noti PC.**

Il prezzo: il più competitivo del mercato in qu... di prestazioni.

Naturalmente anche le MT/85/86 oltre ai t... anche trascinamento a frizione e consentono p... il trattamento del foglio singolo.

20094 Corsico (MI) - Via Borsini, 6
Tel. (02) 4502850/855/860/865/870
Telex 311371 Tally I
00144 Roma - Via M. Peroglio, 15
Tel. (06) 5984723/5984406
10099 San Mauro (TO)
Via Casale, 308 - Tel. (011) 8225171
40050 Monteveglio (BO)
Via Einstein, 5 - Tel. (051) 832508

adverteam

Speed dos da Formula 1

Come spremere il drive e forzare il C-64 per trasferire dati alla velocità di oltre dieci Kbyte al secondo

di Alessandro de Simone

Per capire bene le straordinarie possibilità del prodotto che stiamo per descrivere è necessario spiegare il modo in cui, di norma, vengono trasferiti i dati tra computer e periferica.

Un dato, facente parte di un programma o di una elaborazione, è costituito da un byte che, a sua volta, è formato da otto bit. Per come sono fatti, dunque, sarebbero necessari otto fili elettrici (uno per ciascun bit) per trasferire un byte alla volta.

I computer Commodore, come connessione per periferiche, utilizzano, invece, la trasmissione cosiddetta seriale che consiste nell'inviare (o ricevere) verso (da) la periferica i singoli otto bit costituenti il dato, uno alla volta.

Il motivo di questa scelta è dovuto a motivi economici: un "Bus" (=insieme di connessioni elettriche) costituito da otto fili elettrici è decisamente più difficile da realizzare in modo economico sulla piastra del calcolatore. Quanto asserito trova conferma nel differente prezzo dei connettori per grossi computer che, come è noto, hanno in dotazione un Bus parallelo.

Il grande vantaggio di una economia di produzione si traduce però, inevitabilmente, in una minore velocità di ricetrasmissione.

Per capire il perchè, vediamo come funzionano i due sistemi:

• Collegamento parallelo

Il Bus è costituito, principalmente, da nove cavi. Quando il computer deve inviare un dato, "posiziona" gli otto bit che lo costituiscono sui corrispondenti otto fili elettrici.

Completata questa operazione, il nono filo "comunica" alla periferica interessata che il dato può esser letto. Il modo di segnalare tale possibilità può esser realizzato in molti modi: ad esempio tenendo a "0" lo stato del nono cavo durante il posizionamento del byte e mettendolo a "1" quando l'operazione è conclusa o viceversa.

Oltre ai nove cavi sono presenti altri

segnali che comunicano al computer che il dato è stato correttamente ricevuto dalla periferica e che quindi può esser inviato il successivo.

Tale procedura di scambio di dati si chiama, in inglese, "Handshaking" che, letteralmente, vuol dire "Stringersi la mano". Il termine deriva dal fatto che si immagina che computer e periferica, ogni volta che completano una funzione, si stringano la mano per segnalare, appunto, il termine dell'operazione e la possibilità di effettuare quella successiva.

Ma ciò che più interessa è che la velocità può esser misurata in byte dato che un intero byte viene trasferito nell'unità di tempo.

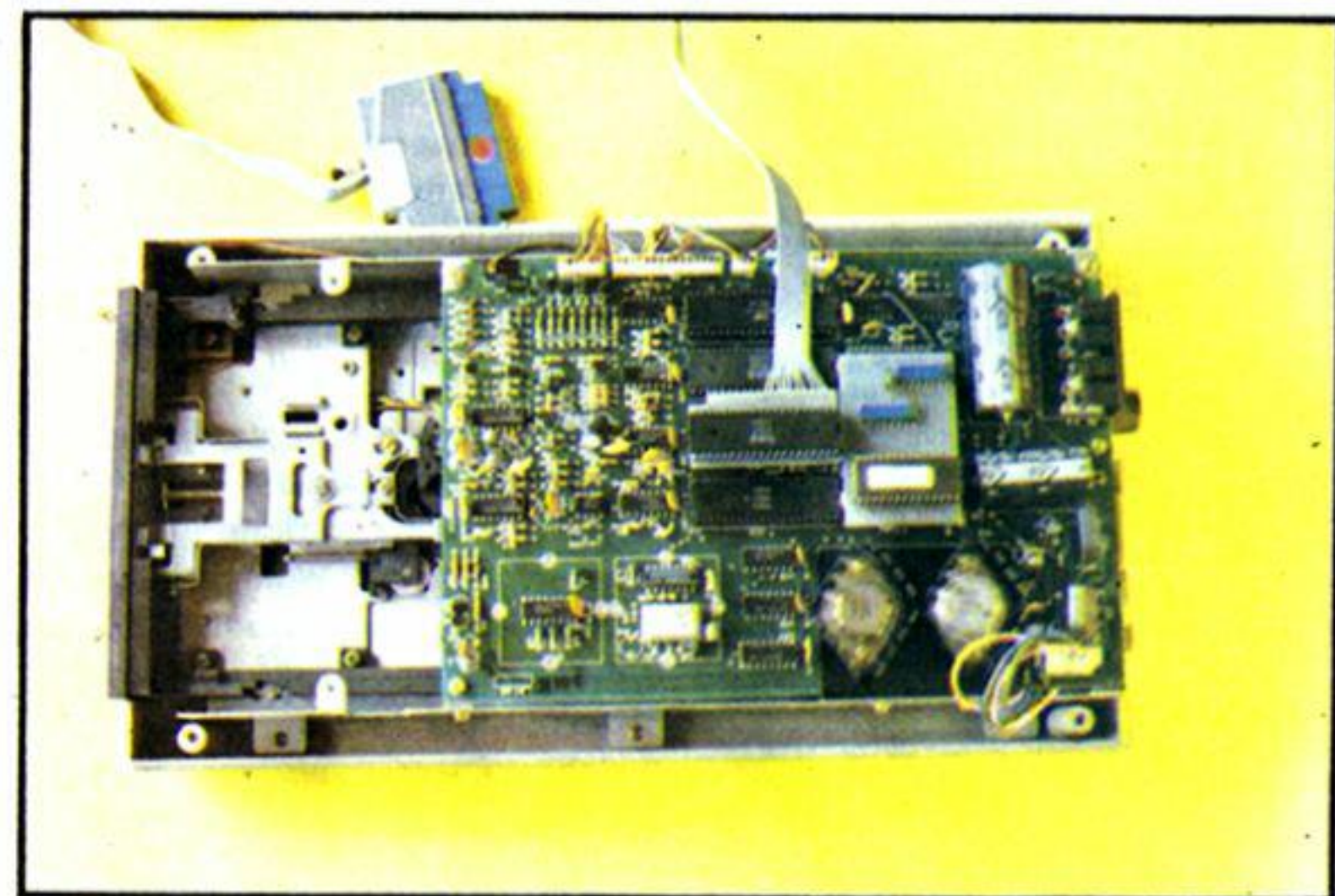
• Collegamento seriale

Con questo tipo di rice—trasmissione, come è intuitivo, i cavi del Bus sono in numero decisamente minore.
Quando un byte deve esser trasferito, il computer immette, sullo stesso cavo, uno alla volta gli otto bit che lo costituiscono. Si suppone, dunque, la presenza di una sorta di contatore software che consente di sapere il momento in cui è stato inviato l'ultimo dato e di prepararsi a ricevere il successivo "pacchetto" di otto bit.

In questo caso la velocità si misura in bit che, come è intuitivo, risulta, agli atti pratici, decisamente inferiore a quella parallela.

Affidabilità

La bassa velocità di trasmissione realizzabile col drive 1541 non è dunque dovuta al desiderio di una maggiore affidabilità ma proprio al fatto che, inviando un bit alla volta, non è prudente elevarla. Se fosse possibile inviare un byte intero, alla stessa velocità con cui si invia un bit, l'affidabilità resterebbe, grossomodo, invariata.

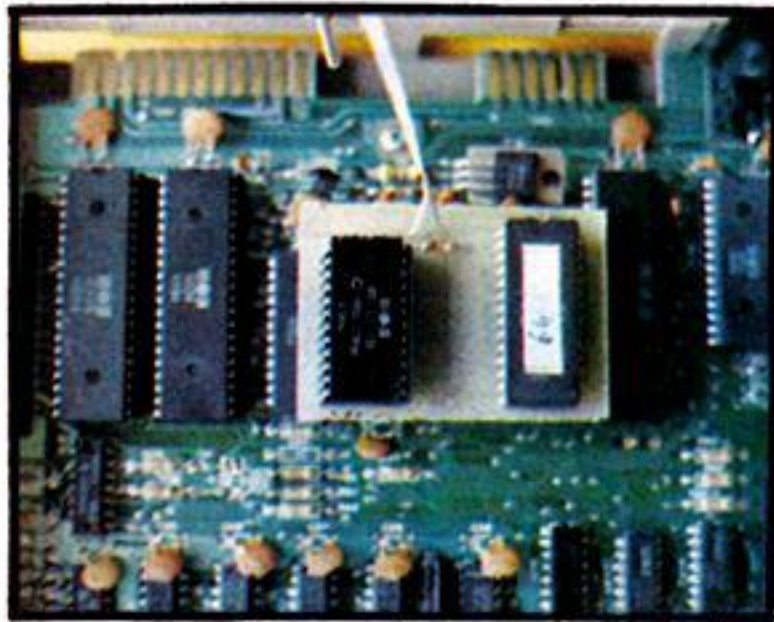
Vi sono, comunque, alcune cartucce che, modificando una parte del Kernal, consentono di aumentare la velocità di invio. Questo, però, rimane di tipo seriale e non elevabile, per questioni di prudenza, oltre un certo limite.

Lo Speed Dos

La Ditta Niwa Soft di Sesto San Giovanni (P.O. Box N.83 Tel.02/2440776) fornisce, al prezzo di L.100000, un sistema hardware—software in grado di raggiungere velocità di trasferimento dati impensabili per un computer della fascia bassa, come, appunto, il C/64.

Come si può notare dalle fotografie, con l'inserimento del "Turbo" sono necessari due cavi tra computer e periferica. Il primo è quello seriale "normale". Il secondo è quello parallelo che collega, tramite la porta utente, l'interno del drive.

Per "truccare" il vostro sistema è necessario effettuare alcune operazioni Hardware che, sebbene semplicissime, richiedono attenzione e precisione in modo da evitare corti circuiti. Per andare incontro anche a coloro che non se la sentono di manomettere il prorio sistema, la Niwa Soft invia, con la confezione, l'elenco dei centri italiani autorizzati a compiere l'operazione.

Ecco, comunque, ciò che si deve fare per dotare il proprio sistema dello Speed Dos:

- Aprire il C/64 e rimuovere dal suo zoccolo la Rom indicata sul manuale. In alcuni esemplari di C/64 (pochi, per fortuna) tale circuito integrato (C.I.) risulta saldato sulla piastra e, di conseguenza, è necessario dissaldarlo per inserire, al suo posto, lo zoccolo.
- Inserire la "vecchia" Rom sullo zoccolo vuoto della prima scheda aggiuntiva (che contiene la nuova Rom alternativa) e installare quest'ultima sullo zoccolo lasciato vuoto precedentemente. Collegata alla scheda è un deviatore che consentirà, a seconda della sua posizione, di utilizzare il nuovo C.I. (Speed Dos) oppure quello "vecchio" che farà funzionare il C/64 nel "solito" modo. La compatibilità col software precedente è quindi totale dato che il nuovo C.I. può essere totalmente escluso e sostituito dall'originale agendo semplicemente sul deviatore.
- Aprire il drive 1541, rimuovere dal suo alloggiamento il circuito integrato indicato sul manuale ed inserirlo sulla seconda scheda aggiuntiva, contenente un'estremità del cavo parallelo. Inserire la scheda nell'alloggiamento lasciato vuoto e richiudere il drive.

Le prove effettuate

Anzitutto è stata provata la compatibilità hardware sia con programmi professionali (Word Pro, Data Base, Spread Sheet) che con videogiochi sofisticati già in nostro possesso da vecchia data (alta risoluzione, sintesi vocale, multi sprite eccetera): tutto ha funzionato regolarmente.

Passati in modo Speed Dos abbiamo constatato con piacere la maggiore velocità di caricamento degli stessi programmi di prima (oltre, ovviamente, al loro corretto funzionamento).

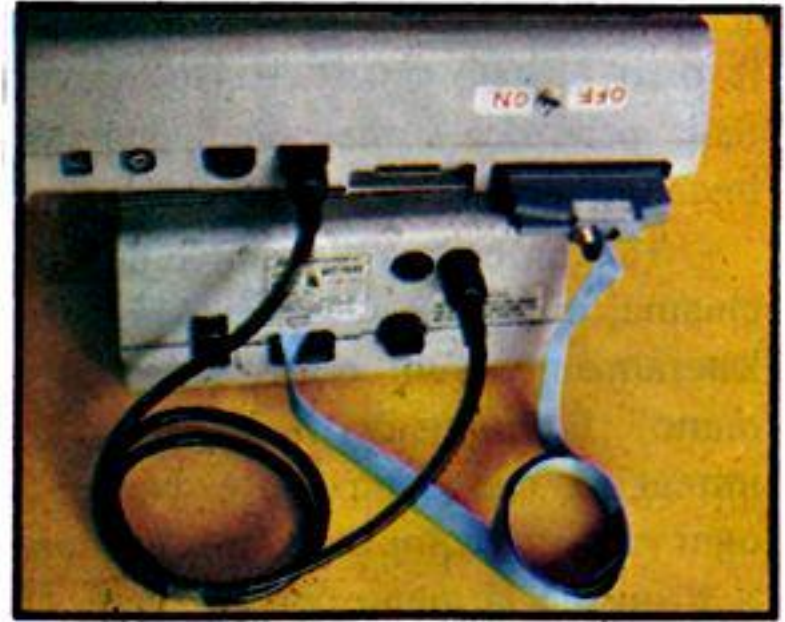
Abbiamo notato, soprattutto, la profonda differenza esistente tra lo Speed Dos e le solite cartucce che consentono di aumentare la velocità.

Queste, in effetti, consentono di aumentare la velocità di caricamento dei programmi ma, una volta "lanciati", la velocità di trasferimento dei dati elaborati rimaneva la stessa.

Il vantaggio di caricare ad alta velocità un programma di Word processor, come l'Easy script, veniva annullato, insomma, dalla impossibilità di "scaricare" su disco, alla stessa velocità, i testi elaborati.

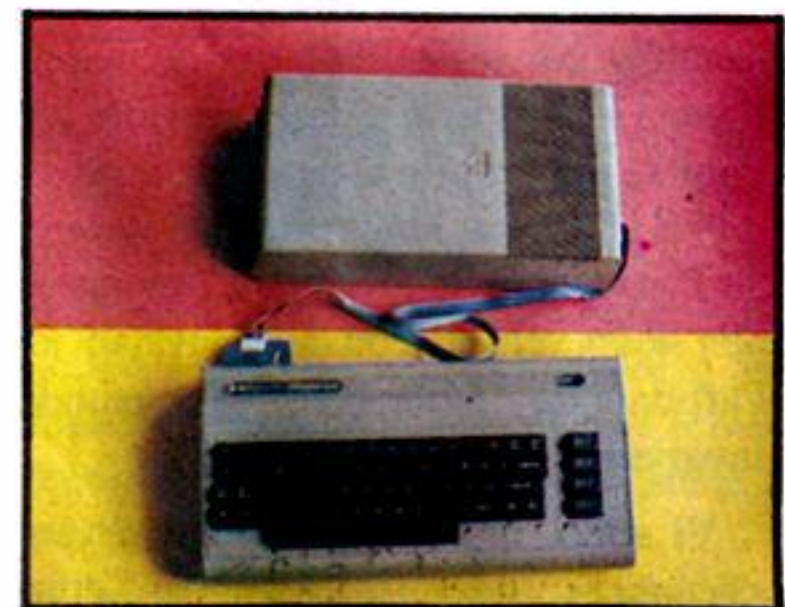
Dalla tabella pubblicata si può invece notare che con lo Speed Dos la velocità rimane alta anche nella gestione dei file sequenziali da e verso disco.

La massima velocità

La ditta Niwa Soft inserisce nella confezione un programma speciale che, utilizzato con lo Speed Dos, consente di aumentare ulteriormente la già alta velocità di trasferimento dati.

Tale software, formattando un disco in modo diverso dal consueto, consente di caricare un programma di 202 blocchi (vedi tabella) in soli cinque secondi! Tenendo conto che gran parte di questo tempo è dedicato, dal sistema, per elevare la velocità del disco dalla immobilità al giusto livello, si può immaginare facilmente con quale impeto vengono scagliati i dati sulla superficie magnetica.

Questo sistema di formattazione consente di realizzare una biblioteca di software Hi—Speed da utilizzare con grande semplicità.

Concludiamo precisando che programmi registrati nel modo descritto possono esser letti solo da sistemi C/64 e 1541 muniti di Speed Dos a causa, appunto, della formattazione speciale cui si è accennato.

Qualsiasi altro disco può invece esser letto e scritto, ad alta velocità, con la certezza di utilizzarlo anche su sistemi "normali".

	TEMPO (secondi)		
N. block	Norm.	Speed Dos	Hi-Speed Dos
105 prg (load)	67	27	4
105 prg (save)	/	55	/
202 prg (load)	127	46	5
202 prg (save)	/	46	/
95 seq (load)	63	24	/
95 seq (save)	80	55	/

Nozioni per l'utente moltiplicate per 9 in una sola opera

L'attuale tecnica dei microcomputer Vi offre:

- **Una documentazione esauriente del hardware con descrizione dettagliata**
dei processori 8085, 8086, 6800, 68000, 6502/6510, Z 80, Z 8000 con gli schemi di collegamento e comandi, la memoria centrale, in più il RAM statico e dinamico, sempre con schemi di collegamento ed estratti delle tabelle dati, delle memorie principali (ROM, PROM, EPROM etc.)
delle interfaccia (RS 232/V24, Centronics Parallelo, IEC-BUS etc.)
dell'equipaggiamento di input (tastiere, joystick, penne ottiche etc.)
- **Un corso dettagliato MC**, che sull'esempio concreto del microcomputer Vi aiuta a comprendere il linguaggio Assembler, i tipi di comandi, i tipi di indirizzamento, gli aiuti di programmazione e le correlazioni.
- **Istruzioni di montaggio compresi i layouts delle platine** etc. per apparecchi supplementari per un computer monoplatine come per esempio una platina al bus ed una platina di input/output.
- **Un corso completo di linguaggio di programmazione BASIC**
- **Programmi applicativi** e simili e problema N-Damen, Quicksort, smistamento binario, Renumber 64, interfaccia Centronics per C 64.
- **Tabelle dati** con indicazioni abbreviate per i microprocessori, circuiti integrati, circuiti periferici.
- **Edizioni supplementari all'opera di base** con nuovi programmi, corsi di linguaggi (fra l'altro PASCAL, ASSEMBLER) istruzioni di costruzione per espansioni di memoria, interfaccia, panoramiche di mercato e indicazioni attuali.

Richiedeteci ancora oggi:
Attuale tecnica di microcomputer
Un raccoglitore robusto in pelle artificiale, formato DIN A4 ca. 450 pagine, numero d'ordinazione 1400 – Prezzo L. 68.000. – Ogni 2–3 mesi riceverete un volume supplementare di completamente all'opera di base con ca. 120 pagine al prezzo di pagina di L. 250.

GARANZIA
Lei ha la possibilità di esaminare i libri in casa. Se deciderà di non essere interessato entro 10 giorni può ritornare il volume ricevendo il suo denaro in cambio, sempre dato che i libri siano in perfetta condizione.

Il programma di tune-up per il vostro Commodore 64/128

Questo manuale sviluppato in particolar modo per il Commodore 64/128 Vi offre:

- **Programmi funzionali al 100% e routine ausiliarie** per l'economia, la tecnica, la grafica ed il suono, fra l'altro nella parte 9 la SUPERMOUSE nonchè programmi adatti per l'analisi dei dati, l'elaborazione dei testi e l'amministrazione di un archivio.
- **Corsi di programmazione riferiti al calcolatore per i linguaggi superiori della programmazione ed Assembler.**
L'opera di base metterà fra l'altro a Vostra disposizione un assemblatore affermato, un disassemblatore ed un monitor linguaggio macchina.
- **Descrizioni dettagliate dei sistemi** con descrizione particolareggiata dei processori (coprocessori), per sound e video chips nonchè moduli di memoria del Vostro modello 64 (128).
- **Ampliamenti ed accessori interessanti** – la parte 7 Vi mostra come potete completare il Vostro 64 con CPM e come Vi potete creare con il software da moduli EPROM un nuovo sistema operativo.
- **Istruzioni complete di costruzione comprese le platine** fra l'altro per un collegamento completo di un IEC (compreso software) e di una penna ottica.
- **Supplementi all'opera di base:**
con nuove routine utili e programmi, corsi intensivi di linguaggi come LOGO, PASCAL e FORTH, nuovi recenti sviluppi e tante altre cose ancora.

Richiedeteci ancora oggi:
Nuove possibilita' con il commodore C 64/128
Un raccoglitore robusto ad anelli formato DIN A4, opera di base con ca. 400 pagine, numero d'ordinazione 2000 – al prezzo di L. 68.000.
Riceverete i supplementi di completamento ogni 2–3 mesi con ca. 120 pagine, al prezzo per pagina di L. 250.

Nuove possibilità con il Commodore 64/128

EDIZIONI WEKA s.r.l.
Via Don Carlo Gnocchi 7
20148 Milano

55 programmi in BASIC per il tempo libero e la professione

I 55 programmi contenuti nell'opera di base sono stati scritti in microsoft BASIC. Delle indicazioni riferite ai particolari specifici delle attrezzature Vi consentono un adeguamento senza problemi anche al Vostro apparecchio, riceverete fra l'altro:

- **Programmi matematici**, inoltre la Regola di Cramer, operazioni a matrici, statistica, calcolo differenziale ed integrale
- **Programmi economici** come la contabilità del valore in contanti, piano di ammortamento di un mutuo, cambio di valuta etc.
- **Programmi tecnici**, per esempio tabella consumo benzina, tabella valori per circuiti digitali,
- **Programmi generici** come aiuti di addestramento professionale, controllo della salute e giochi.

Per ogni programma riceverete una discrezione dei problemi, il modo di procedere, un programma strutturale, un listing e test di prova.

Questa offerta di programmi viene ampliata con supplementi che escono ogni quattro mesi, per adeguarsi continuamente alla ultime novità nel settore del MC.

- **... e per i principianti: Tutto sulla programmazione.**

La parte teoretica Vi guiderà partendo dal giusto modo di procedere nella introduzione delle righe di programma, le costanti e le variabili fino ai comandi ed alle funzioni.

Richiedeteci ancora oggi:
Attuali programmi in BASIC
Un raccoglitore robusto di plastica in formato DIN A4, opera di base ca. 550 pagine, Numero d'ordinazione 1300 – Prezzo L. 68.000.
Riceverete i supplementi di completamento ogni 2–3 mesi con ca. 120 pagine al prezzo di L. 250 per pagina.

☒ Si, speditemi subito

- ☐ **Attuale tecnica di microcomputer** Un raccoglitore robusto in pelle artificiale, formato DIN A4 ca. 450 pagine, numero d'ordinazione 1400 – Prezzo L. 68.000.
- ☐ **Nuove possibilita' con il commodore C 64/128** – Un raccoglitore robusto ad anelli formato DIN A4, opera di base con ca. 400 pagine, numero d'ordinazione 2000 – al prezzo di L. 68.000.
- ☐ **Attuali programmi in BASIC** – Un raccoglitore robusto di plastica in formato DIN A4, opera di base ca. 550 pagine, Numero d'ordinazione 1300 – Prezzo L. 68.000.

Mi invierete ogni 2–3 mesi i supplementi all'opera di base di ca. 120 pagine per volta, al prezzo per pagina di L. 250 (posso disdire l'abbonamento in qualsiasi momento, senza indicarne motivi).

Vi preghiamo di tagliare il tagliando e mandare in busta a Edizioni WEKA s.r.l., Via Don Carlo Gnocchi 7, 20148 Milano

Il mio indirizzo:

Cognome, Nome

Via N.

CAP Città

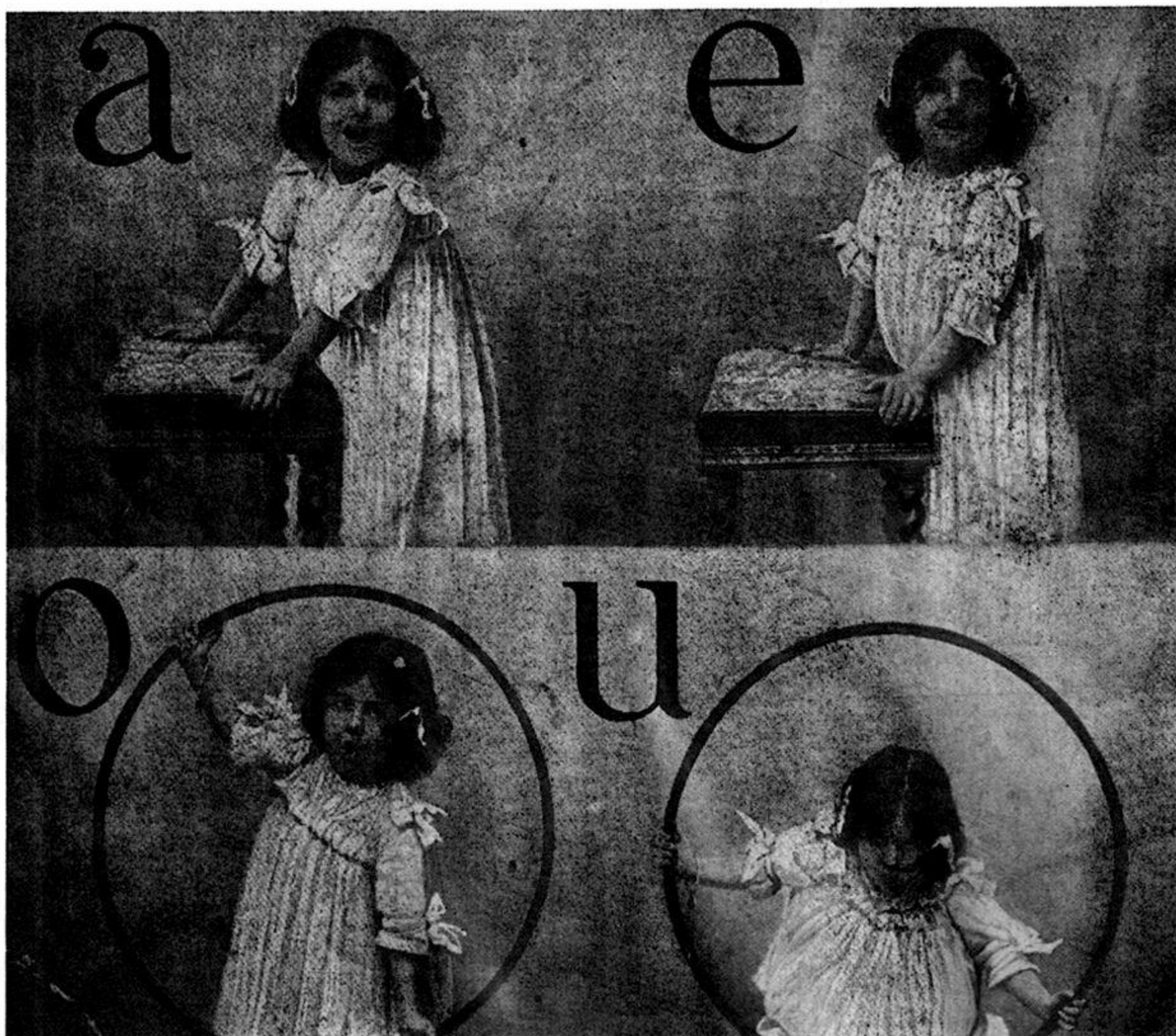
Data Firma

Pago fin d'ora con:

- ☐ assegno non trasferibile intestato a Edizioni WEKA
- ☐ allego Vaglia Postale
- ☐ pagherò quando riceverò il vostro avviso

Discorso

Un prontuario di frasi formate da parti interscambiabili, utili per parlare senza dir... nulla

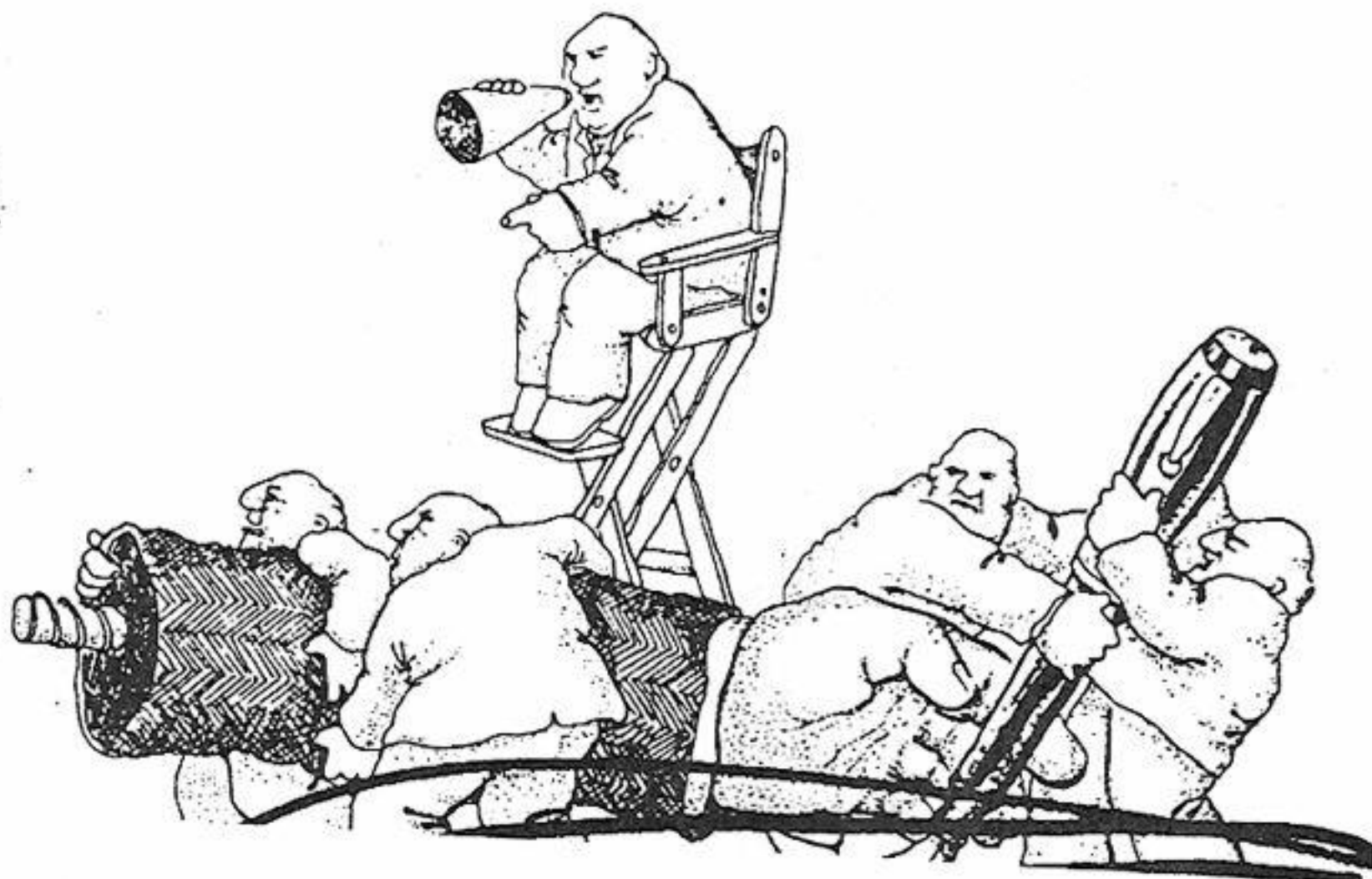
di Michele Maggi

Ecco un programma talmente semplice che non ha certo bisogno di essere commentato nei dettagli.

Si basa sull'uso intensivo dell'istruzione RND (1) che assegna un valore casuale alla variable "I", in funzione della quale sarà visualizzata una frase costruita con i "pezzi" contenuti nei vettori stringa A$, B$, C$, D$, E$.

Qualunque frase ottenuta, anche se a prima vista potrebbe avere un significato, in realtà non vorrà dire assolutamente nulla.

Divertitevi a modificare i "pezzi" di frasi o ad aggiungerne altri: chissà che non scopriate in voi una vena oratoria....

```
 10 REM ****************
 20 REM *   QUALSIASI   *
 30 REM *   COMMODORE   *
 40 REM *               *
 50 REM D I S C O R S O !
 60 REM *               *
 70 REM *(MICHELE MAGGI)*
 80 REM *               *
 90 REM ****************
100 :
110 REM  --- SOGGETTO ---
120 :
130 A$(1)="LA RICONVERSIONE,"
140 A$(2)="L'ATTEGGIAMENTO INDI
    VIDUALISTA,"
150 A$(3)="IL QUALUNQUISMO BURO
    CRATICO,"
160 A$(4)="L'INDUSTRIALIZZAZION
    E PRECOSTITUITA,"
170 A$(5)="LA COMPONENTE RIFORM
    ISTA,"
180 :
190 REM  -- GERUNDIO --
200 :
210 B$(1)="EVIDENZIANDO ASPETTI
     NUOVI,"
220 B$(2)="FUNGENDO DA CATALIZZ
    ATORE,"
230 B$(3)="ALIMENTANDO UNA POLI
    TICA DI ASPETTATIVE,"
240 B$(4)="RISCOPRENDO VECCHIE
    PROBLEMATICHE,"
250 B$(5)="ALTERANDO I MODELLI
    PRECEDENTI,"
260 :
270 REM  -- VERBO --
280 :
290 C$(1)="SI PROPONE DI ATTUAR
    E"
300 C$(2)="TENDE A REALIZZARE"
310 C$(3)="AUSPICA"
320 C$(4)="PERSEGUE"
330 C$(5)="STIMOLA"
340 :
350 REM  -- COMPL. OGGETTO --
360 :
370 D$(1)="UNA COMPLETA DIPENDE
    NZA INTERSETTORIALE"
380 D$(2)="UNA RISTRUTTURAZIONE
     RADICALE"
390 D$(3)="UN POTENZIAMENTO STR
    UTTURALE"
400 D$(4)="UN MIGLIORAMENTO A L
    IVELLO DI SETTORE"
410 D$(5)="UN RINNOVAMENTO DI C
    ARATTERE GENERALE"
420 :
430 REM  -- FINALE --
440 :
450 E$(1)="ASSICURANDONE LA BUO
    NA RIUSCITA."
460 E$(2)="PER IL BENESSERE COL
    LETTIVO."
470 E$(3)="PER LA RISOLUZIONE D
    I OGNI CONTROVERSIA."
480 E$(4)="REALIZZANDO IL PROGE
    TTO INIZIALE."
490 E$(5)="POTENZIANDO LE INFRA
    STRUTTURRE."
500 PRINTCHR$(147):GOSUB 590:PR
    INTA$(I)
510 GOSUB 590:PRINTB$(I)
520 GOSUB 590:PRINTC$(I)
530 GOSUB 590:PRINTD$(I)
540 GOSUB 590:PRINTE$(I)
550 PRINT:PRINT:PRINT:PRINT"ANC
    ORA?(S/N)"
560 GET A$:IF A$="" THEN 560
570 IF A$<>"S" THEN END
580 GOTO 500
590 I=INT(RND(1)*5+1):RETURN
```

Nuovo Sistema

Dopo il successo dell'Enciclopedia di routine in Basic (iniziata col N.24 di C.C.C.), dopo l'introduzione al Linguaggio Macchina, in data più recente, ecco a voi...

di Alessandro de Simone

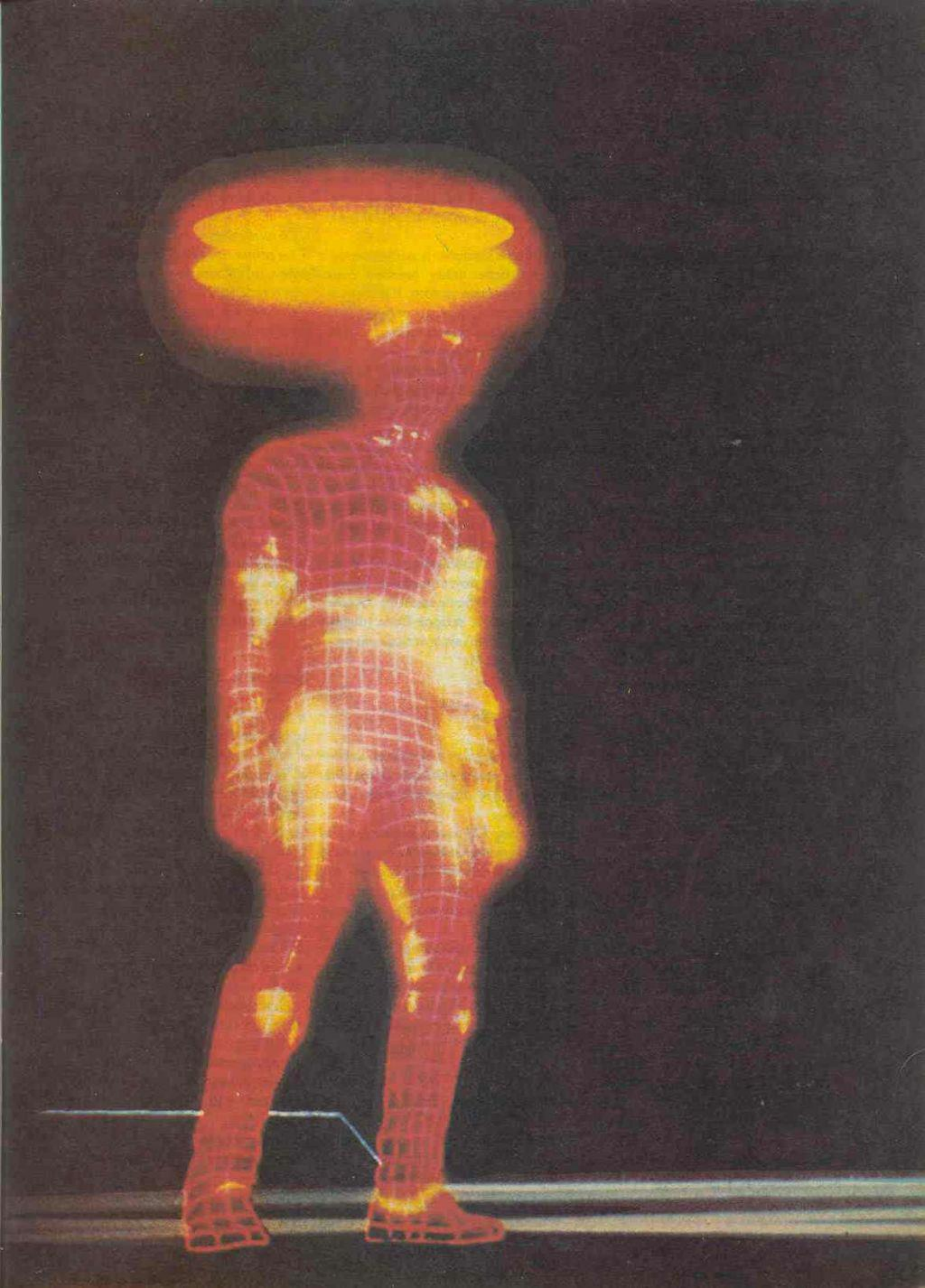
Potevamo farne a meno? Come era possibile interrompere un discorso così proficuo, soprattutto in seguito all'incoraggiamento e alle sollecitazioni dei nostri lettori? Iniziamo pertanto, a partire da questo numero, una nuova "sezione" dedicata sia ai principianti sia a coloro che ritengono di conoscere il Commodore come le proprie tasche (!).

I primi, con un minimo di buona volontà, potranno utilizzare le simpatiche routine pubblicate anche senza sapere minimamente come funzionano: dovranno limitarsi a digitarle e a "lanciarle".

Siamo comunque sicuri che, prima o poi, verranno contagiati dal meraviglioso Virus del Linguaggio Macchina.

Gli altri, gli esperti, avranno a disposizione una stupenda palestra in cui esercitarsi e scoprire che, in fondo, conoscere il computer senza applicare l'esperienza acquisita, è come avere un'automobile in garage e non usarla.

Il Nuovo Sistema

Abbiamo più volte detto che un qualsiasi computer altro non è se non un insieme di mini-istruzioni scritte secondo un linguaggio, il Linguaggio Macchina

appunto, che può esser ben compreso solo dal microprocessore (detto anche micro, per brevità) installato nel computer stesso.

Il Basic (o un altro linguaggio qualsiasi) ed il sistema operativo (S.O.), devono esser considerati solo dei "filtri" interposti tra l'utilizzatore ed il micro al solo scopo di rendere semplicissime operazioni altrimenti mostruosamente complesse.

E' difficile immaginare il tempo dedicato alla progettazione delle ROM del computer per fare in modo che voi, in seguito alla semplice battitura di alcuni tasti, possiate pervenire rapidamente a precisi risultati.

Tanto per fare un esempio, lo studio del "percorso" seguito dal micro, dal momento della digitazione del comando PRINT"PIPPO" alla visualizzazione della parola PIPPO è straordinariamente lungo ed ha richiesto, ai tecnici della Commodore, numerose ore di lavoro, tentativi, prove e ottimizzazioni successive. E tutto questo per rendere l'uso della tastiera semplice al punto da poter essere usata anche da un bambino.

Se, dunque, un'istruzione Basic, un comando, oppure un'operazione qualsiasi sono, in fin dei conti, un aggregato di istruzioni in Linguaggio Macchina, chi ci impedisce di aggiungere altri codici macchina per arricchire le prestazioni del nostro personal?

L'uovo di Colombo

L'idea che stiamo per presentare può far sorridere chi il computer lo possiede da molto tempo. Si sa benissimo che specifici programmi di utilità, in grado di aggiungere nuovi, potenti istruzioni al Basic standard Commodore, sono in circolazione da moltissimo tempo e non rappresentano una novità per nessuno.

Ricordiamo, tra gli altri, il Simon's Basic, il Tool, il Basic 4.0 e perfino le nostre stesse routine grafiche pubblicate tempo fa.

Purtroppo, però, tali programmi devono esser utilizzati globalmente e non è possibile, ad esempio, servirsi contemporaneamente di un comando del Simon's Basic e di un'istruzione del Basic 4.0.

Il motivo è intuibile: la particolare architettura delle utility possono esser spesso in contrasto tra loro. E tutto questo, ovviamente, a parte considerazioni di ordine pratico come la rimozione delle protezioni dei programmi stessi, il loro disassemblaggio per individuare le "zone" relative ai comandi desiderati, la rilocazione e la fusione degli stessi in un unico blocco.

Con la sezione NUOVO SISTEMA intendiamo proporre, appunto, la creazione di routine in L.M. tali che possano esser utilizzate raggruppandole, magari, in più versioni fatte su misura per l'utente e le varie occasioni.

Ogni routine pubblicata in queste pagine, insomma, dovrà soddisfare alcune regole inevitabilmente tassative. Queste riguardano sia la sezione Basic (digitazione dalla rivista) sia la sezione più propriamente in L.M.

Sezione Basic

- *Se risulta necessario, si utilizza, dapprima, il programma che limita il Top della memoria RAM disponibile (programma Top Memo).*
- *Le Routine LM vanno caricate grazie al breve programma standard (denominato, appunto "Caricatore") che contiene tutti gli accorgimenti utili per allocare correttamente i codici macchina delle singole routine.*
- *In seguito all'allocazione, è possibile registrare la zona RAM in cui risiede la routine LM, grazie ad un accorgimento presente nello stesso "Caricatore" oppure mediante l'apposito programma "Save Zona RAM".*

Sezione READ... DATA

Tenendo conto di quanto detto nei punti precedenti, le righe Basic (da aggiungere al "Caricatore") che contengono la sezione LM vera e propria, devono possedere le seguenti caratteristiche:

- *La numerazione inizia col valore 1000.*
- *La prima riga (1000) deve avere, come prima istruzione, la cancellazione dello schermo: PRINTCHR$(147).*
- *Le righe che seguono la N.1000 devono contenere, sotto forma di PRINT, tutte le istruzioni idonee al corretto utilizzo della routine LM*
- *La riga successiva all'ultimo comando PRINT delle istruzioni precedenti è costituita dall'unico comando RETURN.*
- *Seguono, quindi, le righe che, grazie ad un'istruzione DATA, contengono i codici macchina decimali relativi alla routine LM.*
- *L'ultima riga DATA deve esser costituita da due soli valori: —1 (che ha il compito di indicare al programma "Caricatore" la fine dei codici) ed il valore Cheksum (che si incarica di controllare l'esatta trascrizione della routine).*

Sezione L.M.

- *Le routine devono esser rilocabili, nel senso che l'utente può decidere la zona RAM più idonea alla loro allocazione.*
- *Devono, di conseguenza, funzionare perfettamente in modo del tutto indipendente dalla loro collocazione fisica all'interno della memoria RAM scelta dall'utente.*
- *Devono essere, per quanto possibile, indipendenti da routine del Sistema Operativo e garantire, comunque, un corretto funzionamento nel caso sia indispensabile riferirsi al S.O.*
- *Non possono far ricorso a tecniche che alterino il vettore dell'Interrupt.*
- *Non possono far ricorso a tecniche che alterino il contenuto dello stack.*
- *Non possono far ricorso a tecniche che alterino puntatori vitali del Basic.*
- *Non devono richiedere alterazioni di puntatori o di vettori per il loro caricamento.*
- *La prima istruzione della routine deve essere quella cui far riferimento con la SYS.*

... e non indurci nei Syntax Error, e liberaci dagli Out of Data...

Come interpretare correttamente alcuni messaggi di errore emessi dal vostro Commodore, e per non arrabbiarsi mai più.

Avvertenza

Numerose lettere di nuovi lettori, che si dichiarano in preda al panico per i continui errori segnalati dal proprio Personal, ci hanno indotto a scrivere un intero articolo che accomunasse le spiegazioni necessarie per venirne a capo.

Descriveremo nei minimi dettagli, come al solito, i casi più frequenti in modo da rendere chiaro il motivo della generazione dei messaggi stessi da parte del calcolatore.

E' doveroso precisare, da parte nostra, che una parte del presente scritto è sostanzialmente identica all'articolo "Gli errori di Sintassi", pubblicato, in analoghe circostanze, sul N.18 di C.C.C. La decisione di riproporlo è stata presa per conferire la dovuta completezza ad un argomento ritenuto basilare.

Nel presente testo, per motivi di brevità, si farà ricorso alla simbologia qui di seguito indicata:

- *Tasto Crsr Destra (= premere il tasto che consente di spostare il cursore a destra)*
- *Tasto Crsr Sinistra; Tasto Crsr Alto; Tasto Crsr Basso (di analogo significato)*
- *(R): Premere il tasto Return*

Attenzione, inoltre, a non confondere il tasto zero "0" col carattere alfabetico "O".

Due errori molto, molto comuni

Siete pronti? Bene, accendete il computer e incominciamo a... sbagliare.

Stiamo parlando sul serio: nulla è più proficuo, nell'uso di un

calcolatore, che sbagliare a bella posta per individuare più facilmente, in futuro, il motivo delle segnalazioni d'errore.

Se avete appena acceso il vostro Commodore (qualunque esso sia) dovreste veder lampeggiare il cursore al di sotto della parola READY. Digitate la seguente istruzione:
PRINT 2 +2 (R)

Se avete fatto esattamente come vi abbiamo suggerito, dovrebbe esser comparso il risultato dell'operazione (4) ed il cursore, che è riapparso di nuovo sotto la "R" di READY, tornerà a lampeggiare.
Ora seguite *alla lettera* queste fasi:

- *Premete una sola volta il tasto Ins/Del: Il cursore, in conseguenza di ciò, si sposta all'estrema destra del rigo su cui si trova il messaggio "Ready".*
- *Premete una sola volta la barra spaziatrice. A questo punto il cursore è tornato nello stesso punto di prima (al di sotto di "Ready").*
- *Digitate lo stesso ordine di prima (PRINT2 +2) e premete il tasto Return.... Sorpresa! Viene emesso il messaggio Out of Data Error. Che diavolo è successo?*

Per spiegarci questo strano fenomeno, dobbiamo risalire al modo in cui viene gestito, nei computer Commodore, il rigo del video su cui digitiamo i dati che, successivamente, devono esser elaborati dal computer.

Prendete un qualsiasi listato Basic, caricatelo sul vostro Commodore, digitate List (R), ed attendete che compaia l'ultima riga. Se siamo fortunati (!), il listato dovrebbe comparire assortito: righe di comandi Basic che occupano un solo rigo di schermo dovrebbero coesistere con righe Basic che ne occupano due.

Se, purtroppo, il programma che avete caricato possiede solo righe Basic costituite da un solo rigo di video (o soltanto righe Basic di due righe video), caricate altri programmi in vostro pos-

sesso oppure digitatene uno costituito da linee contenenti soltanto istruzioni REM seguite da tanti caratteri quanti ne bastano per realizzare righe Basic lunghe una sola oppure due righe di video.

Supponendo, dunque, che sul vostro schermo sia visualizzato un programma "assortito", premete (imprimendo una sola pressione alla volta) il tasto che comanda il movimento del cursore in basso. Dovreste riuscire a notare che, (a patto che lo premiate per un solo breve istante alla volta) quando il cursore giunge in fondo al video, alla successiva pressione del tasto Crsr una parte del listato scompare verso l'alto a causa del noto fenomeno dello "Scrolling" (=scorrimento).

Ripetendo più volte l'esperimento noterete che lo Scrolling è di una sola riga se, prima della pressione del tasto Crsr, sulla sommità del video è presente una riga Basic lunga un solo rigo di schermo.

Al contrario lo scrolling risulterà di due righe se la prima riga Basic visibile in alto sullo schermo è costituita da due righe video.

In altre parole lo scrolling viene effettuato in modo tale da evitare la permanenza di "mezze" righe di listato Basic. Ciò vuol dire che siamo in presenza di un sistema di scorrimento intelligente, capace, cioè, di riconoscere la lunghezza e il "tipo" di ciò che viene visualizzato sullo schermo in quel particolare momento.

Per ribadire il concetto, effettuate il seguente esperimento:

- *Spegnete e riaccendete il computer (oppure servitevi del tasto di Reset, se lo possedete).*
- *Digitate il seguente microprogramma ricordando di premere il tasto Return al termine della digitazione di ciascuna delle due righe:*

100 REM RIGA UNO
110 REM RIGA DUE

- *Posizionate il cursore (mediante i tasti Shift e Crsr Destra) sul carattere "O" della parola "UNO" della riga 100.*

• *Tenente premuto il tasto Crsr Destra: Noterete che il cursore, giunto all'estremità della riga Basic (che coincide, in questo caso, con la riga video su cui è posizionato), riappare sul rigo in basso che, lampeggiando, supererà. Sembra che non sia successo nulla e, in effetti, è proprio così...*

• *Posizionate ora il cursore nuovamente alla destra della "O" di "UNO" di riga 100 e premete, stavolta, la barra spaziatrice invece del tasto Crsr Destra. Noterete che il cursore, giunto all'estrema destra del rigo, passerà, come prima, al rigo successivo ma, fatto importante, la riga 110 viene spostata in basso di una posizione!*

Chi possiede il C-16 si accorgerà che, addirittura, non riuscirà mai a raggiungere la riga 110 tenendo premuta la barra, perchè tutte le volte che il cursore raggiunge il bordo destro del video, la riga 110 viene spostata in basso di una riga.

Col C-64 (e col Vic-20), invece, lo scroll avviene solo una volta: tenendo premuta la barra spaziatrice, prima o poi riuscirete a raggiungere la riga Basic successiva.

Che cosa c'entra tutto questo col punto dal quale eravamo partiti?

E' giunto il momento di svelare il mistero.

Quando, al momento dell'accensione, compaiono i "soliti" messaggi (Commodore... Basic... Bytes free... eccetera), il computer non sa (nè può saperlo) il reale significato di ciò che è visualizzato sullo schermo in quel momento.

"Egli" è fermamente convinto che ogni riga (oppure "doppia" riga) del video rappresenti un gruppo di istruzioni da eseguire immediatamente oppure appartenga ad un programma Basic da memorizzare per future elaborazioni.

Quando, dunque, posizionate il cursore alla destra della parola READY, e premiamo il tasto Return, il computer è indotto ad elaborare ciò che è presente sul rigo. La parola Ready viene però "interpretata" come "READ" "Y" e, dato che non vi sono istru-

zioni DATA, viene emesso l'idoneo messaggio "Out Of Data Error".

I risultati del primo esperimento suggerito (quello del PRINT2 +2, tanto per intenderci) sono ora facilmente comprensibili:

- *Portandoci col cursore all'estremità del rigo contenente la parola READY, e premendo la barra spaziatrice, non facciamo altro che comunicare al computer l'intenzione di "legare" la riga video successiva alla precedente. Se, dunque, digitiamo a questo punto il comando PRINT2 +2, e premiamo (R), il computer, poveretto, si vede costretto ad elaborare tutto ciò che ritiene costituisca la riga Basic su cui è posizionato il cursore al momento della pressione del tasto Return. Naturalmente, con questa convinzione, elaborerà per primo il comando "READ" "Y" con le conseguenze già viste.*

La morale che si può trarre è la seguente:

- *Spostandosi col tasto cursore non si cambia la "sostanza" dei caratteri su cui ci spostiamo.*
- *Premendo la barra spaziatrice si corre il rischio di "fondere" più righe successive senza che noi ce ne accorgiamo.*

Un errore analogo al precedente

Spesso molti lettori si ostinano a definire errati alcuni programmi pubblicati perchè, secondo loro, vengono emessi Syntax Error durante l'elaborazione nonostante i listati siano trascritti fedelmente dalla nostra rivista.

Poichè (credeteci) controlliamo più volte i programmi, prima di pubblicarli, siamo rimasti per lungo tempo perplessi di fronte alle incrollabili certezze dei... contestatori.

Finalmente, un bel giorno, venne a trovarci in Redazione un lettore munito di un nostro listato registrato (da lui) su nastro per dimostrarci che l'errore di sintassi c'era, eccome.

Naturalmente l'errore c'era, ma era da attribuire ad una errata digitazione da parte dell'ostinato lettore. E veniamo al dunque cercando di spiegare in che modo è possibile generare errori inspiegabili.

- *Spegnete e riaccendete il computer oppure premete il tasto di Reset.*
- *Cancellate lo schermo (Shift e Clr/Home).*
- *Digitate ALLA LETTERA la seguente riga (e NON premete il tasto Return finchè non ve lo diciamo noi!)*

100 A=1234567890:B=1234567890:A$="PIPPO"

Poichè è molto importante per la comprensione dell'errore che stiamo per commettere, precisiamo che il carattere "1" del numero di riga (100) deve essere il primo carattere della riga video. Vi deve essere un solo spazio tra il numero 100 e la definizione della variabile A=123 eccetera. Non vi deve essere alcuno spazio tra le assegnazioni successive.

Se, insomma, avete fatto **esattamente** *come vi abbiamo detto, a questo punto delle operazioni il cursore dovrebbe lampeggiare sulla prima cella video posta al di sotto del carattere "1" del numero di riga 100.*

- *Scrivete ora:*

110 PRINT A,B,A$

e, finalmente, premete il tasto Return.

- *Cancellate lo schermo e digitate List (R).*

Il listato **sembra** costituito da due righe (100 e 110) ma, in effetti, è costituito da un'unica riga dato che, giunti alla fine "logica" della riga 100, **non** avete premuto il tasto Return.

Il listato, così come appare sul video, sembra privo di difetti e ci

aspetteremmo che, digitando RUN (R), compaiano i contenuti delle due variabili (A, B) ed il nome PIPPO.

Col Run, al contrario, viene visualizzato un dannato "Syntax Error in 100".

Morale:

Quando compare una segnalazione di errore di sintassi (o di altro tipo) riscontrata in una certa riga, "chiedete" il List relativo **soltanto** a quella riga e **non** ad un gruppo di linee (oppure, addirittura, all'intero listato). Se, infatti, nel caso appena esaminato digitate List 100 (R) dovreste accorgervi che qualcosa non va dato che voi chiedete una sola riga ed il computer (sembra) darvene due.

Putroppo molti lettori, specie se principianti, digitano semplicemente List (R) e, almeno nei casi come quelli visti, non riescono ad accorgersi che alcune linee visualizzate sono, in realtà, un'unica linea Basic.

Ricordiamo che è possibile incorrere nell'errore descritto posizionandosi col cursore sull'ultimo carattere di una riga Basic e tentando di posizionarsi sul rigo successivo ricorrendo alla pressione della barra spaziatrice. In questo caso, come abbiamo già visto all'inizio, provochiamo involontariamente la "fusione" di due righe.

In conclusione:

- *Premete SEMPRE il tasto Return al termine di una riga Basic, anche se sembra che non ve ne sia bisogno.*
- *Se il computer segnala un errore in una determinata linea, chiedete il listato relativo SOLTANTO alla linea segnalata in modo da evitare equivoci.*
- *Se volete posizionarvi rapidamente all'inizio della riga successiva (senza alcuna altra conseguenza) premete insieme i tasti Shift e Return.*

Ma ecco altre cause d'errori.

che possano generare uno degli errori più comuni programmando in Basic.

L'errore di Sintassi

L'errore più comune che il principiante commette è sicuramente l'errore di sintassi: SYNTAX ERROR.

E' bene rendersi conto che il Basic è un linguaggio del tutto simile al nostro almeno per ciò che riguarda le regole di sintassi, di grammatica eccetera. La differenza consiste nel fatto che il computer è rigoroso, spietato e pignolo. Basta un minimo errore perchè il calcolatore si blocchi irrimediabilmente. Il cervello umano, e la nostra educazione, consentono di comprendere i messaggi che ci rivolgono pur se contengono errori:
... "Vadi a destra e, al primo semaforo, vadi sempre dritto"...
... "Se avrei fatto una dormita, ora fossi più riposato"
... ed altre, sono frasi che abbiamo spesso sentito e, nonostante tutto, perfettamente compreso. Un computer che sentisse parlare come negli esempi riportati, si scandalizzerebbe immediatamente e comunicherebbe di aver incontrato difficoltà di comprensione. Purtroppo il calcolatore emette un semplice messaggio (SYNTAX ERROR IN XXX) senza indicare esattamente il posto della riga Basic in cui ha riscontrato l'errore segnalato nè, tantomeno, il motivo.

Chi possiede un C-16 oppure un Plus/4, beato lui, ha a disposizione il tasto Help che fa lampeggiare la "zona" interessata dall'errore.

Vedremo ora in che modo rintracciarlo all'interno di una linea di programma. Continueremo a commettere errori a bella posta e ad effettuare le dovute correzioni fino a che tutto risulti in ordine. Ricorreremo, come di consueto, a numerosi esempi che, meglio di un qualsiasi fiume di parole, renderanno chiaro il concetto di

errore di sintassi.

Digitate il seguente programma dopo aver spento e riacceso il computer (e non dimenticate il tasto Return al termine di ciascuna riga...):

```
100 A=123: B=A*C: C=10
110 PRINT;A PRINT;B PRINT;C
```

Approfitteremo di questo listato anche per evidenziare in seguito alcuni errori di logica. Il problema, che dovrebbe esser risolto dal programma proposto, consiste nell'eseguire un prodotto tra due valori ("A" e "C") e visualizzarlo. C'è da ricordare che il calcolatore esegue le istruzioni l'una dopo l'altra a cominciare dalla prima incontrata. Nel nostro caso particolare non appena digitiamo RUN (ed il tasto Return), il computer esamina il contenuto della riga con numerazione più bassa (100 nel nostro caso). La prima istruzione di tale riga associa alla variabile "A" il valore 123. Subito dopo (notare la presenza del carattere di doppio punto che separa le due istruzioni), si chiede di eseguire il prodotto di A*C e di associarlo alla variabile B. Benchè l'operazione sia corretta, il programma non soddisfa il problema perchè, a questo punto delle operazioni, la variabile "C" contiene il valore nullo (0) dato che ogni volta che si digita RUN tutte le variabili sono automaticamente poste a zero. A nulla vale il fatto che la terza istruzione di riga 100 ponga C=10.

Tentando di far partire il programma, otteniamo un Syntax error in 110 (che d'ora in poi indicheremo semplicemente con S.E.) Ciò è dovuto al fatto che tra le istruzioni PRINT è indispensabile inserire il carattere di doppio punto (:).

Come si può notare, il messaggio visualizzato dal computer indica anche la riga del programma Basic in cui è stato riscontrato l'errore.

A questo punto chiedete il listato (List 110) e, servendovi dei tasti di controllo, "salite" col cursore, posizionatelo nei posti

appropriati e battete il carattere di doppio punto ove necessario. Non dimenticate di battere il tasto Return alla fine dell'operazione e, per maggior sicurezza, chiedete nuovamente il listato per verificare che il calcolatore abbia accettato le modifiche imposte.

Digitando RUN non dovrebbero, ora, esservi inconvenienti di sorta ad eccezione dell'errore "logico". Il vostro Commodore, infatti, stamperà, l'uno sotto l'altro, i valori 123, 0, 10 mentre il problema richiedeva come soluzione i valori 123, 1230, 10.

Per ottenere una tale risposta è necessario modificare la riga 100 come segue:
100 A=123:C=10:B=A*B

Considerazioni sull'esperienza effettuata

- *Tra due istruzioni è indispensabile inserire il carattere doppio punto (:)*
- *Il computer non può in alcun modo individuare errori logici di programmazione. L'unico modo per individuare errori di questo tipo consiste nel far "girare" il programma e nel verificare i risultati con altri già noti.*
- *Tra due caratteri di doppio punto (:) deve figurare un'istruzione (oppure un comando) purchè COMPLETI.*

Esempio:
*100::PRINT A,B,C::PRINT A*B*C*
(corretto)
*100:PRINT:A:B:C:PRINT:A*B:*C*
(ERRATO)

- *Il numero di doppi punti tra due istruzioni successive non costituisce alcun... pericolo come pure il numero di spazi tra due istruzioni o trà gli elementi di una stessa istruzione. Provate a modificare il programma precedente come segue:*

*100::: A = 1 23::: C =10 :: B=A*C*

L'unica differenza consiste nel fatto che il programma occupa più spazio in memoria e che il listato è di più difficile lettura.

L'uso improprio del doppio punto però, in alcuni casi, facilita la lettura del listato pur se a prezzo di una maggior occupazione di memoria. La stessa riga di prima, ad esempio, si può spostare sulla destra (notare il doppio punto subito dopo la numerazione di riga):

*100: A=123: C=10: B=A*B*

Tale comodità può esser apprezzata maggiormente nel caso in cui si disponga di una stampante.

Un altro errore logico

Rimanendo fermi sullo stesso esempio visto all'inizio, cercheremo ora di rintracciare altri errori "logici" che nessun computer è in grado di individuare. In alcuni casi non è obbligatorio l'uso del punto e virgola. Modificate ad esempio la riga 110 come segue:

110 PRINT A: PRINT B: PRINT;C

Il risultato che otterrete digitando RUN (e Return) dimostra che:

- *Dopo la parola-comando PRINT non è indispensabile il carattere di punto e virgola.*
- *Se non si inserisce quest'ultimo carattere, il computer interpreta PRINTA come se fosse presente il punto e virgola (PRINT;A).*
- *Non è indispensabile inserire uno spazio tra l'istruzione e la virgola (PRINTA produce lo stesso effetto di PRINT A).*

Ciò premesso sembrerebbe che il punto e virgola sia un carattere superfluo. Vedremo ora perchè non è così.

Modifichiamo ora il programma come segue:

100 A=123:C=10:B=A*B

110 PRINT A;:PRINT C;:PRINT B

Grazie alla presenza dei caratteri di punto e virgola otterremo, dopo il RUN, i tre valori affiancati anzichè l'uno sotto l'altro. Allo scopo di "risparmiare" caratteri sul listato, possiamo modificare la riga 110 come segue:

```
110 PRINT A;C;B
```

Battendo RUN e Return constatiamo, infatti, che il risultato è identico al precedente. Poichè abbiamo detto che il carattere di punto e virgola è superfluo, ci sentiamo autorizzati a digitare la riga 110 come segue:

```
110 PRINT A B C
```

Otteniamo, invece dei tre risultati di prima, un semplice zero (0) e nient'altro. Come mai?

I computer Commodore permettono di utilizzare variabili definite con uno oppure con due caratteri, come ad esempio A=12 oppure AC=12345 oppure A4=67. Variabili definite con più di due caratteri sono egualmente accettate ma memorizzate utilizzando i soli due primi caratteri.

Per esempio digitare NU=40 oppure NUMERO=40 o, ancora, NUOTARE=40 è la stessa cosa. Il calcolatore, in altre parole, associerà alla variabile "NU" qualsiasi valore, calcolo o elaborazione che interesserà NU comunque venga chiamato (NU, NUM, NUMERO, NUMISANTI eccetera).

Ritornando al listato precedente, quando il computer incontra

```
PRINT A B C
```

ritiene di dover visualizzare la variabile "ABC" che, diversa da "A", "B" oppure "C", vale zero come qualsiasi altra variabile non dichiarata esplicitamente. Ecco dunque che l'istruzione:

```
PRINT A;B;C
```

è profondamente diversa da:

```
PRINT A B C
```

proprio per la presenza del carattere di punto e virgola. Volendo utilizzare variabili con nomi lunghi, ecco un sistema per farlo correttamente:

```
PRINT NUMERO; AREA; IPOTENUSA
```

che equivale a:
PRINT NU;AR;IP
ed è diverso, lo ripetiamo, da PRINT NUARIP

La comprensione del listato, ovviamente, è migliore nel primo caso ed ha l'unico inconveniente di occupare più memoria.

Conclusioni sull'esperienza

- *Non sempre il punto e virgola è indispensabile. Talvolta è possibile sottintenderlo.*
- *Il punto e virgola (e, analogamente, la virgola) è indispensabile tra variabili diverse. Omettendolo possono sorgere difficoltà di interpretazione da parte del computer e portare ad effettuare elaborazioni non desiderate benchè prive di errori.*

Le variabili "vietate"

Abbiamo detto che è possibile utilizzare variabili indicandole con due caratteri alfabetici e, in certi casi, alfanumerici. Ce ne possiamo render conto ricorrendo ai seguenti esempi:
AA%=182: B1=-43.5: CR=3*45-B1 eccetera

Non tutte le combinazioni sono però permesse. Se provate, in modo diretto oppure sotto programma, a far eseguire:
LENTE=18
otterrete un S.E. Il motivo è presto detto: LENTE è una variabile formata da tre caratteri alfabetici utilizzati anche per la funzione LEN(X$). Questa ha il compito di "restituire" la lunghezza della stringa X$ (tratteremo più avanti questo argomento).

Il computer, interpretando carattere per carattere, incontra dapprima "L", subito dopo "E" e infine "N". A questo punto si aspetta di incontrare il carattere di parentesi aperta (come, ap-

punto, richiede la corretta sintassi).

Dato che ciò non avviene (incontra infatti il carattere "N"), emette la segnalazione di errore (S.E.)

Analogamente altre variabili con nomi di fantasia possono emettere E.S. se, al loro interno, contengono nomi di funzioni o comandi Basic. Esempi:

LENTO, CATETO (TO)
OGGETTO, ESEGETA (GET)
MAREMMA, PREMIO (REM)
MASCHIO, ASCETA (ASC)

Altre variabili, i cui caratteri, d'altra parte, non coincidono con quelli iniziali di comandi o istruzioni, sono egualmente "vietate". ST è infatti riservata al Sistema Operativo (O.S.) del calcolatore per memorizzare eventuali condizioni di errore nel colloquio con il registratore e il drive. ST è, come si suol dire, una variabile a sola lettura: è possibile esaminare il contenuto ma non definirlo. In altre parole è lecito PRINT ST ma è illecito ST=46. Un'altra variabile a sola lettura è TI, utilizzata dal calcolatore per comunicare i dati relativi all'orologio interno. TI$, invece, non è vietata, è cioè alterabile a piacere dall'utente a patto che la stringa sia lunga sei caratteri. Per esempio, il comando:
TI$="111007"
"regola" le ore 11, 10 minuti e sette secondi. TI$="11" genera, al contrario, un "Illegal Quantity Error" perchè i caratteri non sono sei.

Se state utilizzando particolari programmi di utility, come il Basic 4.0, oppure i più recenti computer C-16, Plus/4, C-128, altre variabili possono esser considerate vietate. In questi casi, infatti, le variabili DS e DS$ sono variabili a sola lettura.

Il comando PRINT

Come è noto, i calcolatori Commodore accettano, nella digita-

zione, abbreviazioni dei comandi. Per esempio invece di END si possono digitare, di seguito, i caratteri "E" ed il tasto "N" battuto insieme col tasto SHIFT. Invece di PRINT, inoltre, è sufficiente battere il punto interrogativo (?) eccetera. Siccome è possibile sbagliare nella digitazione delle abbreviazioni, il computer, incontrando un'istruzione "strana", emette il messaggio S.E. In questi casi è sufficiente visualizzare la riga indicata nel messaggio e controllarla attentamente. Esempio:

Se il computer visualizza SYNTAX ERROR IN 430
digitate LIST 430 ed esaminate la riga.

In un solo caso l'errore non è individuabile e ciò si verifica quando si ricorre all'abbreviazione "?" nella battitura del comando PRINT # tipico delle operazioni di scrittura dati su periferiche (registratore, drive, stampante, ecc.). L'abbreviazione cui si accenna è infatti illegale: è necessario battere per esteso PRINT oppure ricorrere all'abbreviazione "P" (battuto normalmente) e tasto "R" (battuto insieme con SHIFT).

Conclusioni sull'esperienza

- *Tutte le variabili (numeriche e stringa) non possono avere "incorporati" gruppi di caratteri che costituiscono le parole "chiave" di comandi o istruzioni Basic.*
- *TI, ST ed altri nomi sono egualmente vietati (variabili a sola lettura).*
- *TI$ deve sempre essere costituita da sei caratteri.*
- *PRINT # non può essere abbreviato con ?# .*

Gli errori relativi a vettori e matrici

Dovreste ricordarvi che, allo scopo di memorizzare valori nu-

merici, è possibile ricorrere non solo a variabili numeriche di tipo intero (esempi: AA%=45: B1%=28: X%=-105) o a variabili decimali (AA=45.67: B1=-28.54: X=.789) ma anche a variabili individuate da un numero particolare detto "indice".

Nell'esempio che segue è riportato un breve programma applicativo. Digitatelo dopo aver spento e riacceso il calcolatore.

```
100  DIM X(12)
110 FOR I=0TO 12
120  X(I)=I
130  PRINT X(I)
140  NEXT
```

Il programma non fa altro che inserire, nel vettore X(I), il valore "I" al posto che compete. Entrambi si incrementano in modo eguale grazie al ciclo FOR... NEXT.

Spieghiamoci meglio: La riga 100 "genera", nella memoria del computer, uno spazio sufficiente per memorizzare tredici valori individuati da un indice (numero, cioè, racchiuso tra parentesi) che può variare da zero a 12 per un totale, appunto, di tredici (e non 12) "caselle". E' come se avessimo creato una tabella simile a quella della schedine del totocalcio con la differenza che, invece di poter scrivere in ciascuna di esse soltanto simboli "1", "2" oppure "X" possiamo memorizzare un qualsiasi valore intero o decimale, positivo o negativo.

Le righe 110... 140 eseguono un conteggio da zero a 12 e, a mano a mano che questo prosegue, viene associato a ciascuno dei tredici elementi il valore stesso del conteggio: nella "casella" numerata con zero sarà memorizzato il valore nullo, in quella numerata con "1" il valore unitario e così via.

Il listato, se digitato correttamente, funziona a dovere e visualizza, dopo aver digitato RUN (R), i tredici valori. Provate ora a cancellare la riga 100.

Ricordiamo che per cancellare una riga è sufficiente battere il

numero della riga stessa (100, in questo caso) e premere il tasto Return. Alcuni principianti ritengono, erroneamente, che una riga di programma venga eliminata cancellandola interamente mediante la pressione della barra spaziatrice. Con quest'ultima operazione, infatti, non succede assolutamente nulla e una conferma del fatto che la linea esiste ancora la potete ottenere digitando List (R) ed osservando ciò che visualizza lo schermo.

Accertatevi, dunque, che la riga 100 sia effettivamente stata cancellata chiedendo il listato (List): dovrebbero apparire solo le righe da 110 a 120.

Fate partire il programma con RUN (R): verranno visualizzati 11 valori (da 0 a 10) ed il messaggio "Bad Subscript Error in 120".

Se provate a listare la riga in questione (List 120) vi accorgerete che non vi sono errori di sintassi. L'errore riscontrato dal calcolatore, infatti, consiste nel fatto che, se un vettore non è dimensionato esplicitamente (cioè è privo del comando DIM), il numero dei suoi elementi viene sempre e comunque fissato in 11 (numerati da 0 a 10). Il motivo della segnalazione di errore è dovuto al fatto che, quando il conteggio del ciclo FOR... NEXT giunge a 11, il computer non può rintracciare nella sua memoria la dodicesima "casella" e blocca, di conseguenza, l'elaborazione. Se, infatti, dopo l'interruzione del programma, proviamo a chiedere il valore di "I" (PRINT I), possiamo notare che il computer risponde con 11.

Cancellate ora il programma con NEW (R), e battete il seguente:

```
100 DIM X(100)
110 INPUT"QUALE CASELLA";N
120 INPUT"DIGITA NUMERO";V
130 X(N)=V: GOTO 110
```

La riga 100 dimensiona un vettore di 101 elementi (numerati da 0 a 100), definito con il carattere alfabetico "X". La riga 110

chede il numero della "casella" in cui vogliamo trascrivere il valore che digiteremo al momento dell'INPUT di riga 120.

Trascritto il programma, ecco un esempio di come può essere utilizzato:

RUN (R)
Quale casella ? 53 (R)
Digita numero? -65.98 (R)
Quale casella? 12 (R)
Digita numero? 9868.67 (R)
eccetera.

Nel modo appena visto, nella 53ma "casella" (che chiameremo, più propriamente: elemento) del vettore "X", sarà presente il valore -65.98, nella 12ma sarà contenuto 9868.67 e così via. Questo programma, per come è strutturato, non serve assolutamente a nulla se non per individuare casi particolari che possano generare errori.

Alla domanda "Quale casella?" provate, infatti, a rispondere con un numero maggiore di 100 oppure con un numero negativo. In entrambi i casi il programma si blocca emettendo un nuovo tipo di errore:

"Quale casella?" 678 (R)
"Digita numero?" 6 (R)
?Bad Subscript error in 130
Ready.

Si noti che il messaggio di errore non si riferisce alla riga 110, ma alla 130. In questa, infatti, compare l'istruzione X(N)=V. Elaborando la riga 110 il calcolatore non può sapere in che modo sarà utilizzato il valore che noi digitiamo al momento dell'INPUT e si limita semplicemenete ad accettarlo associandolo alla variabile "N". Battendo PRINT N, dopo l'interruzione del programma,

possiamo renderci conto del motivo per cui il comando non è stato eseguito. In casi come questo è sufficiente rintracciare la riga in cui compare DIMX() e verificare, appunto, che l'indice di dimensionamento è inferiore a quello che abbiamo cercato di far accettare.

Modifichiamo ora il programma aggiungendo le seguenti righe senza cancellare le altre:

```
115 IF N<0 OR N>100 THEN GOSUB 200: GOTO 110
200 FOR I=0 TO 100
210 PRINT I"ELEMENTO="X(I)
220 NEXT:RETURN
```

Col programma modificato in tal modo, digitando un valore negativo o superiore a 100, alla domanda "Quale casella?", l'elaborazione prosegue alla subroutine 200-220 e, subito dopo, alla riga 110 (vedi Goto 110 di riga 115)

La subroutine 200-220 visualizza (riga 210) il numero di ciascuno dei 101 elementi del vettore "X", il messaggio "Elemento=" ed il contenuto dello stesso.

In definitiva la subroutine 200-220 farà apparire sul video i 101 messaggi richiesti. Esempio:

```
RUN (R)
"Quale casella?" 98 (R)
"Digita numero?"  5 (R)
"Quale casella?" 99 (R)
"Digita valore?"  7 (R)
"Quale casella?" 456 (R)
0 Elemento=0
1 Elemento=1
..........
 97 Elemento=0
 98 Elemento=5
 99 Elemento=7
```

```
100 Elemento=0
"Quale casella?"
eccetera
```

In corrispondenza di ciascuna casella, insomma, sarà indicato l'eventuale valore digitato in precedenza.

In caso contrario, l'elemento del vettore conterrà sempre il valore nullo.

Si ricorda che se, alla domanda "Quale casella?" si risponde con un valore decimale (come: 55.37) il computer, giunto alla riga 130, troncherà automaticamente la parte decimale e porrà "N" eguale a 55. Si fa notare che, tra l'altro, il programma esaminato può essere considerato come idea base per sviluppare programmi di archivi più complessi.

Cancellate ora il programma (spegnendo e riaccendendo il calcolatore) e digitate il seguente:

```
100 DIM A(3)
110 FOR I=0 TO 10
120 PRINT I;A(I): NEXT
```

Facendolo girare (RUN), però, si ottiene la segnalazione di errore alla quarta visualizzazione (Bad Subscript Error).

Conclusioni sulla precedente esperienza

- *Un vettore, se non dimensionato esplicitamente, come, ad esempio DIM A(187), viene automaticamente dimensionato col valore massimo di 11 elementi (numerati, però, da 0 a 10).*
- *Un vettore, una volta dimensionato, non può accettare indici superiori a quello già dichiarato.*

• *Se un vettore viene esplicitamente dichiarato con un valore inferiore a 10, non usufruisce del dimensionamento automatico che consente il trattamento di almeno 11 elementi, ma solo di quelli dichiarati.*

• *L'errore "Bad Subscript Error" non ha nulla a che fare col Syntax Error. Per individuare l'errore chiedete (servendosi dell'istruzione PRINT) il valore della variabile indice: è molto probabile che un'errata elaborazione di quest'ultima variabile, precedentemente effettuata, abbia portato il valore dell'indice oltre il limite consentito.*

Cancellate il programma in memoria (NEW) e digitate questo di una sola riga:

100 DIM X(25): DIM X(20)

Quando battete RUN viene visualizzato un nuovo errore ("Redim'd Array Error in 100"). Questo dimostra che, una volta dimensionato un vettore, non è possibile modificarne il numero di elementi.

Digitate ora e fate girare questa riga:

100 DIM X(25): CLR: DIM X(20)

Questa riga, al contrario, non genera alcuna segnalazione di errore. L'istruzione CLR, infatti, cancella tutte le variabili fino a quel momento definite, compresi vettori e matrici.

Attenzione, quindi, ad usare l'istruzione CLR con oculatezza per non perdere dati elaborati fino a quel momento.

Digitate ora:

100 DIM X(10000)

Digitando RUN si ottiene un nuovo messaggio: "Out Of Memory Error". Il motivo risiede nel fatto che, dimensionando un vettore, il computer cerca di definire uno spazio nella memoria a sua disposizione. Poichè ogni elemento di un vettore occupa cinque byte (=locazione di memoria RAM) oltre ad alcuni byte sui quali non ci intratteniamo, il calcolatore dovrebbe avere a sua disposizione oltre 50000 byte di memoria. Il Commodore 64 ne ha circa 39000 mentre il Vic-20, senza espansioni, soltanto 3555. Oltre

allo spazio per i vettori dovrebbe inoltre esserci posto anche per altre variabili numeriche e stringa e, non dimentichiamolo, per lo stesso programma Basic!

Conclusioni sull'esperienza effettuata

• *Non è possibile dimensionare più di una volta un vettore o una matrice, pena la segnalazione "Redim'd Array Error".*
• *Vettori troppo grandi possono superare lo spazio di memoria disponibile con l'emissione di un "Out Of Memory Error".*

L'Illegal quantity error

Questo tipo di errore si riferisce a valori numerici illegali, vale a dire a valori numerici che non possono essere elaborati dal computer perchè in opposizione alla logica matematica oppure alle capacità stesse di elaborazione del computer.

Esamineremo i vari casi iniziando dal più semplice. Tutti sappiamo che la radice quadrata di un numero "X" è data da quel particolare valore che, elevato al quadrato (oppure moltiplicato per se stesso), genera il numero "X".

Ad esempio la radice quadrata di 16 vale 4 perchè 4x4 (oppure 4 elevato al quadrato) fornisce, appunto, il risultato 16. I numeri di cui si desidera "estrarre" la radice, però, devono essere solo positivi perchè non esiste alcun valore che, elevato al quadrato, fornisca un numero negativo. Ad esempio non è possibile calcolare la radice quadrata di "-16" perchè sia "4" che "-4" elevati al quadrato, o moltiplicati per se stessi, generano il risultato positivo 16. Ricordiamo che, infatti, meno per meno "fa" più.

Provate a far girare il seguente programma dopo aver cancellato quello presente in memoria:

```
100 INPUT"VALORE";X
110 PRINT"RADICE=" SQR(X)
120 GOTO 100
```

Se, dopo aver battuto RUN (R) digitate valori positivi in risposta alla domanda "Valore?" di riga 100, il calcolatore risponderà visualizzando (PRINT di riga 110) le corrispondenti radici quadrate. Il valore più piccolo che è possibile digitare è zero.

Se provate a digitare un qualsiasi valore negativo (come, ad esempio, -16), l'elaborazione si blocca con l'emissione del messaggio "Illegal Quantity Error in 110". Anche in questo caso la sintassi del comando è corretta. Ciò che genera il messaggio non è il comando PRINT oppure SQR, ma proprio il contenuto della parentesi (detto ARGOMENTO dell'istruzione), vale a dire il valore associato alla variabile "X". Se, infatti, dopo l'interruzione dovuta all'errore, chiedete PRINT X, noterete che viene, appunto, visualizzato un valore negativo.

L'elaborazione, in altre parole, si interrompe perchè il programmatore (cioè voi) non ha contemplato tutti i casi possibili. Se volete evitare l'interruzione di un programma in casi come questo, dovete inserire, all'interno del programma, adeguate istruzioni che, PRIMA di "pericolose" elaborazioni, decidano se è il caso di effettuarle o meno.

Aggiungete, ad esempio, la riga che segue, al programma già presente in memoria:

```
105 IF X<0 THEN PRINT "ERRORE": GOTO 100
```

Questa riga consente di esaminare il valore di "X" da voi digitato al momento dell'elaborazione della riga 100. Solo se il valore è maggiore o uguale a zero viene effettuato il calcolo, altrimenti compare la parola "Errore" e viene posta nuovamente (Goto 100) la domanda.

Le istruzioni di controllo del tipo IF...THEN sono utilissime se usate con accortezza e costituiscono preziosi "filtri" molto efficaci che impediscono errori indesiderati e, soprattutto, l'interruzio-

ne dei calcoli.

Sarebbe opportuno che il lettore si abitui ad inserire i filtri PRIMA di ciascuna istruzione (o riga Basic) che contenga la possibilità di cadere in errore.

Esaminiamo ora altri casi in cui è possibile che si verifichi un errore.

CHR$(X), SPC(X): L'argomento "X" deve esser compreso tra 0 e 255.
PEEK(X): "X" deve variare tra 0 e 65535
GOTO X: L'indirizzo deve essere un valore intero compreso tra 0 e 63999.
POKE X,Y: "X" può variare tra 0 e 65535 mentre "Y" è limitato all'intervallo 0-255.

Errori nel trattamento di variabili numeriche intere, decimali, stringhe.

Un altro errore commesso comunemente dai principianti è quello di associare a variabili numeriche, stringhe alfanumeriche e viceversa.

Partiamo come al solito, dai casi più elementari. Cancellate il programma presente in memoria e digitate il seguente:

```
100 INPUT A
110 PRINT A
120 GOTO 100
```

Questo microprogramma non fa altro che chiedere un valore numerico (riga 100), visualizzarlo sullo schermo (riga 110) e ricominciare il ciclo (riga 120). Se non avete commesso errori di trascrizione (e sarebbe assurdo per un programma così semplice!) battete, ad ogni apparizione del punto di domanda (?) valori a caso, decimali, interi, positivi o negativi che siano. Se tutto è in ordine, gli stessi valori verranno stampati sullo schermo subito dopo. Consideriamo ora alcuni casi particolari sui quali vale la

pena soffermarsi. Quando compare il punto interrogativo, provate a digitare valori molto grandi come, ad esempio: 10000000000 (dieci miliardi). Vi accorgerete che il calcolatore "traduce" automaticamente il valore da voi digitato nella cosiddetta "notazione esponenziale": 1E +10. Tale simbologia significa: "1" seguito da ("E +") dieci zeri ("10"). Provate a digitare altri valori (come quelli riportati nella colonna di sinistra della tabella che segue) che il computer visualizzerà come nella corrispondente colonna di destra.

10000000000	1E10
-10000000000	-1E +10
0.000000001	1E-10
-0.000000001	-1E-10

Una trattazione approfondita sul calcolo di numeri espressi con la notazione esponenziale sarebbe fuori luogo. Il lettore che volesse approfondire l'argomento troverà esaurienti risposte in un qualsiasi libro di matematica in uso nelle scuole superiori.

Provate ora a digitare un numero enorme come, ad esempio, 1E +99. Otterrete un messaggio di "Overflow error in 100" che indica l'impossibilità, da parte del computer, di elaborare valori così elevati.

Sempre utilizzando lo stesso programma, provate ora a digitare 1E +1 che, "tradotto" significa 10 (1 seguito da un solo zero). Vi accorgerete che il computer non lo visualizzerà come lo avete digitato voi, ma lo stamperà come un "normale" 10.

Considerazioni sull'ultima esperienza

- *Valori troppo grandi o troppo piccoli generano un Overflow*

error.

- *Valori formati da un numero superiore alle nove cifre, come 1234567890, vengono automaticamente "tradotti", in fase di visualizzazione, nella notazione esponenziale (1.23456789 E +09).*

- *Valori per i quali sono sufficienti nove cifre per esser rappresentati "normalmente" (come 1.234 E +3) sono visualizzati in notazione decimale (1234).*

Provate ora a digitare, avendo in memoria ancora lo stesso microprogramma finora esaminato, valori numerici "strani" come, ad esempio, con due virgole (carattere questo che, dovreste saperlo, deve esser sostituito da un punto...). Esempio: 123.456.789. Prendete nota di ciò che accade e tenetelo ben presente nella stesura dei vostri programmi.

Battete un valore al cui interno sia presente il carattere della virgola (e non del punto) oppure un carattere di doppio punto. Il computer risponderà con un messaggio finora sconosciuto: "Extra Ignored".

Ciò vuol dire che il calcolatore non ha accettato (e quindi ha ignorato) gli "altri" (extra) caratteri digitati dopo il carattere di virgola o di doppio punto. Esempio:

```
RUN (R)
? 123:456
Extra Ignored
123
? 789,456
Extra Ignored
789
```

Il computer, come si può notare, ha accettato il valore numerico fornito dalle cifre digitate PRIMA del carattere di doppio punto o di virgola. Se, invece del punto, battete il carattere di punto e virgola

(esempio: 123;456) verrà visualizzato un nuovo messaggio: "Redo From Start".

Questo stesso messaggio compare anche se digitate caratteri alfabetici, come "QWERTY". In quest'ultimo caso il computer non può associare alla variabile numerica "A" di riga 100 il valore (...ma quale valore, del resto?) QWERTY e risponde, di conseguenza, "Redo From Start" (Rifai dall'inizio). Il computer, insomma, ripete la domanda di prima finchè non vi deciderete a battere soltanto caratteri numerici. Notiamo, però, che al contrario dei casi precedenti, i due messaggi di errore ("Extra Ignored" e "Redo From Start") non interrompono il programma.

Cancellate il programma di prima e digitate il seguente:

```
100 INPUT A,B,C
110 PRINT"A="A
120 PRINT"B="B
130 PRINT"C="C
140 GOTO 100
```

In questo caso il computer aspetta che voi battiate tre valori separati tra loro da una virgola. Esempio:

```
RUN (R)
? 1,2,3
A= 1
B÷ 2
C= 3
? 34,67,-45
A= 34
B= 67
C= -45
```

Se, invece di battere tre valori separati dalla virgola, ne battete

uno solo, questo verrà associato alla prima variabile ("A") e, subito dopo, comparirà un doppio punto di domanda (??) per ricordare all'utilizzatore che il computer attende la digitazione di altri valori. Esempio:

? 45
?? 23
?? 78
A=45
B=23
C=78

Conclusioni sull'esperienza effettuata

• *Con il comando INPUT relativa all'immissione di dati numerici, l'elaborazione si blocca fino a che non si digitano tutti i valori richiesti dal comando.*
• *Digitando valori separati da virgola o da doppio punto, il programma non si interrompe ma compare il massaggio "Extra Ignored" per indicare che alcuni dati digitati non sono stati presi in considerazione.*
• *Digitando caratteri alfabetici invece che numerici il computer non accetta il dato e, pur non interrompendo l'elaborazione, ripete la domanda dopo aver emesso il messaggio "Redo From Start".*

Cancellate ora il programma presente in memoria e digitate il seguente:

```
100 INPUT A%
110 PRINT A%
```

```
120 GOTO 100
```

Questo programma si differenzia dal precedente perchè si riferisce a variabili intere (A%) invece che decimali. Se, infatti, dopo il RUN (R) digitate valori interi compresi tra -32768 e +32767, questi verranno visualizzati sullo schermo secondo una modalità analoga a quella di prima (valori minori o maggiori generano un "Illegal Quantity Error"). Se, invece, cercate di digitare valori decimali come, ad esempio, -123.65 (attenzione a digitare il carattere punto e non quello della virgola) vi accorgerete che saranno accettati, ma solo per la loro parte intera e nessun messaggio di errore viene visualizzato.

Modificate ora il programma come segue:

```
100 INPUT A$
110 PRINT A$
120 GOTO 100
```

Questo, senza dubbio, è molto più versatile dei precedenti. Infatti proprio perchè l'istruzione INPUT si riferisce ad una variabile stringa (A$), qualsiasi carattere (eccetto la virgola ed il doppio punto) viene accettato e, grazie alla riga 110, visualizzato. Provate infatti a digitare gruppi di caratteri, cioè stringhe, alfabetiche, numeriche oppure semi-grafiche purchè non più lunghe di due righe video.

Da come abbiamo esposto le cose, però, sembrerebbe che non ci sia verso di far accettare al calcolatore nè il carattere di doppio punto (:) nè quello di virgola (,) pena l'emissione del messaggio "Extra Ignored".

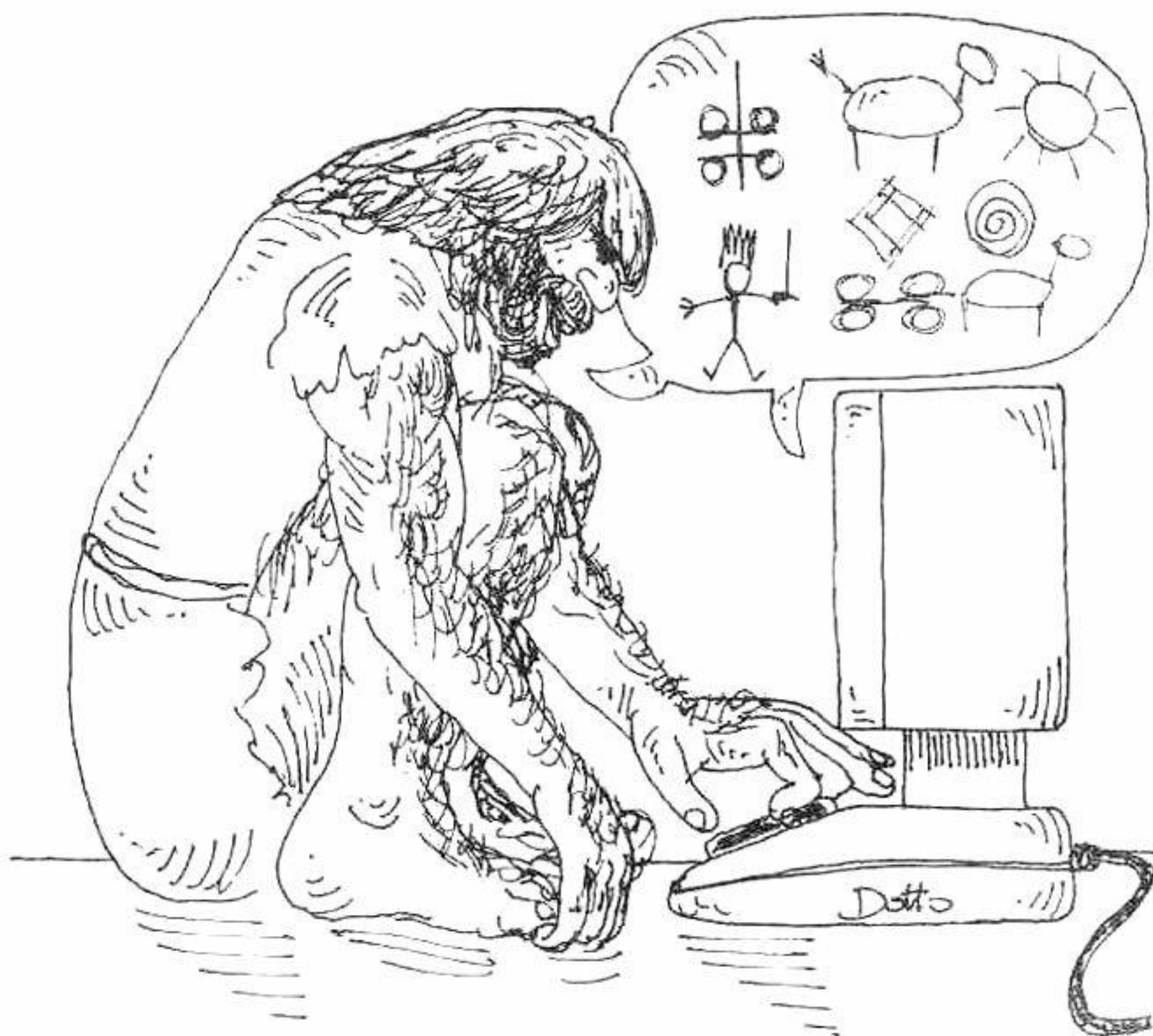
C'è però un semplice artificio per "costringere" il computer ad accettare anche questi caratteri oltre a quelli "speciali" come i caratteri del colore e del controllo del cursore.

Il trucco, dicevamo, consiste nel digitare, come primo carattere della risposta, quello delle virgolette (o apici) che si ottiene pre-

mendo il tasto Shift e il tasto "2". Provate a battere:

```
RUN (R)
? "PROVA:UNO,DUE
PROVA:UNO,DUE
```

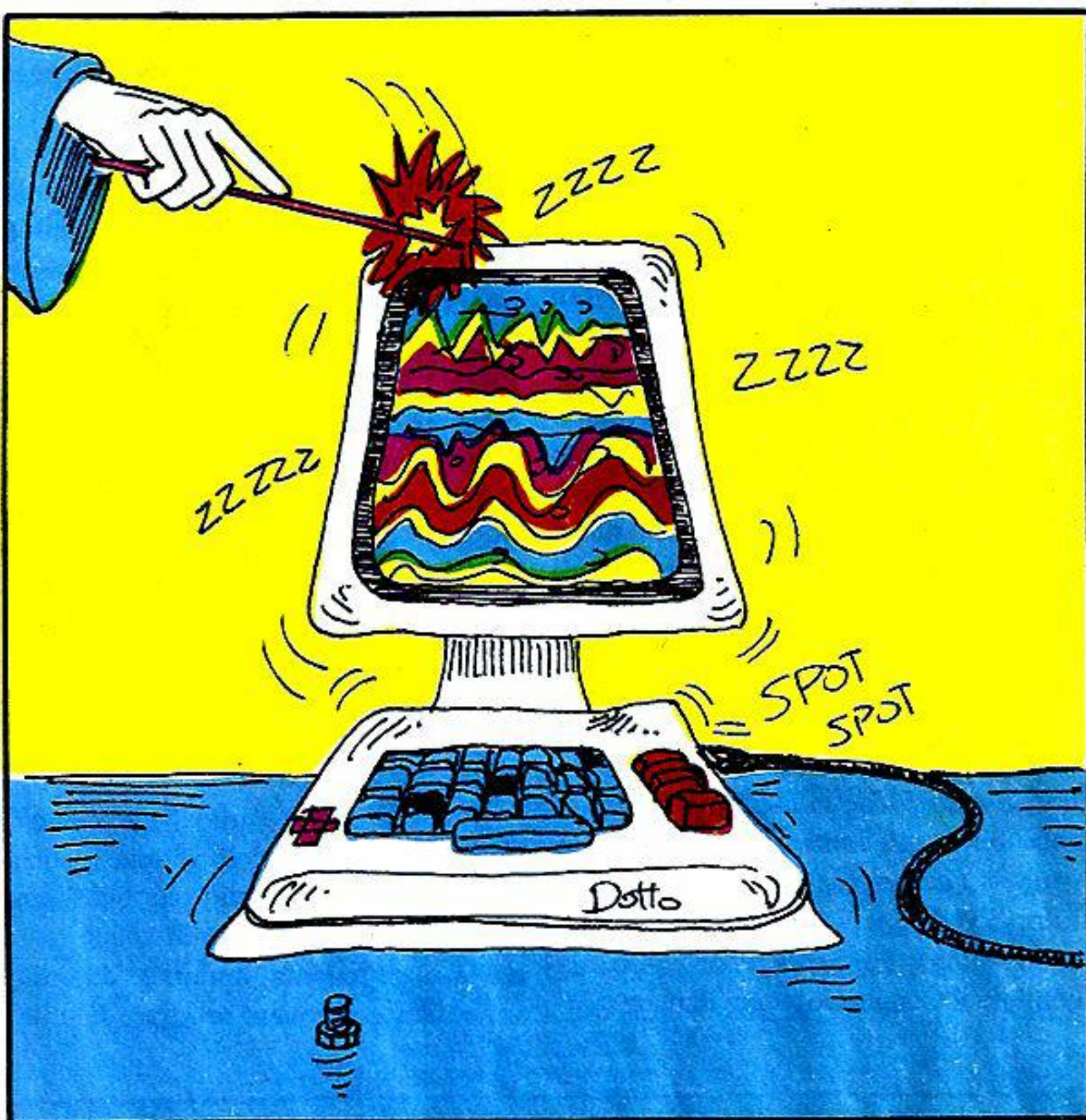
Allo stesso sistema (digitazione delle virgolette) si può ricorrere nel caso si desideri che i primi caratteri della stampa siano degli spazi (i cosiddetti Blank ottenuti premendo la barra spaziatrice).

ΦΨΟΚΛΠΕ
ΩΘΑΓΑΣ
ΑΖΠΥΘΔΦ
ΣΓΖ
Dotto

• *Non possono far riferimento (nè con JSR nè con JMP) ad altre routine che, volta per volta, verranno pubblicate in queste pagine.*

Un esempio pratico

Supponiamo che il lettore trovi interessante la routine "GOTO CALCOLATO" e che voglia utilizzarla anche in propri programmi Basic.

Ecco le fasi da compiere, più lunghe a dirsi che da eseguire:

1 — Accendere il computer (!)
2 — Se questa è la prima volta che utilizzate le routine LM, saltate alla fase N.5.
3 — Caricate, mediante la sintassi LOAD"NUOVO SIS/LM",8,1 (disco) oppure LOAD"NUOVO SIS/LM",1,1 (nastro), il gruppo di routine denominate "NUOVO SIS/LM" che voi stessi, volta per volta, realizzate ed ampliate mediante la fase N.13.
4 - Subito dopo il caricamento, digitare NEW (e il tasto Return), in modo da mettere "a posto" i puntatori Basic per le operazioni successive.
5 — Desiderando allocare la nuova routine "GOTO CALCOLATO" in una zona RAM successiva all'area destinata al Basic, saltate alla fase N. 7.
6 — Caricate e lanciate il programma "Top Memo" per limitare l'area destinata al Basic.
7 — Caricate il programma "Caricatore" (oppure digitatelo dalla rivista).
8 — Digitate, copiandole dalla rivista, le righe Basic del programma "GOTO CALCOLATO" che, se avete seguito a puntino le istruzioni, si "fonderanno" al programma "Caricatore".
9 — Registrate normalmente su disco o nastro, e verificate, il programma Basic così ottenuto.
10 — Digitate RUN ed indicate, quando richiesto, la locazione di partenza in cui si desidera trascrivere la routine LM. Nel caso abbiate già in memoria (fase N.3) una zona RAM contenente routine in LM, prestate la massima attenzione a non creare pericolose "sovrapposizioni". Consultate, in questi casi, il foglio di appunti (Mappa della memoria) da voi scritto durante la fase N.11 effettuata la volta precedente.
11 — Annotate su un foglio di carta (che avrà la funzione di "Mappa della memoria" del NUOVO SISTEMA) la locazione iniziale, quella finale e tutte le informazioni utili che sono comparse sullo schermo. Nel caso sia già presente in memoria un gruppo di routine LM, abbiate l'accortezza di indicare, come indirizzo di partenza, la locazione SUCCESSIVA all'ultima locazione della volta precedente.
12 — Per verificare il funzionamento della routine, caricate il programma DEMO ad essa relativo, riportato sulla rivista, ed attivarlo.
13 — Se siete soddisfatti del funzionamento, e se volete arricchire il vostro NUOVO SISTEMA della nuova routine, caricate il programma "Save Zona RAM" ed indicate la prima e l'ultima locazione della zona da registrare predisponendo l'apposito disco (o nastro) ed aggiungendo sul foglio le nuove indicazioni necessarie per il corretto utilizzo della nuova routine "GOTO CALCOLATO". Il nome da assegnare al file potrà, ovviamente, essere qualunque. Ad esso, comunque, faremo riferimento indicandolo col nome: "NUOVO SIS/LM".

Come utilizzare il programma "Top Memo"

Per facilitare, soprattutto ai lettori principianti, l'utilizzo delle routine LM che verranno di volta in volta riportate su queste pagine, indichiamo qui di seguito tutte le operazioni da compiere per realizzare un NUOVO SISTEMA che definiamo, per ora, "standard" ma che, una volta compreso il meccanismo per arricchirlo sempre di più, potrà essere abban-

donato e sostituito da altri, personalizzati a seconda delle vostre specifiche esigenze.

La funzione Fre(0)

Poichè, per motivi che non stiamo qui a dire, il Sistema Operativo dei computer Commodore utilizza, nel gestire le stringhe, TUTTA la memoria RAM disponibile da Basic, può succedere che l'utente trascriva una routine L.M. in una zona di memoria RAM ritenuta a torto libera e si accorga (spesso quando è troppo tardi per porre rimedio) che il S.O. ha cancellato, magari solo in parte, la routine stessa.

Per evitare che ciò avvenga è indispensabile trarre in inganno il computer comunicandogli che la memoria RAM disponibile è più limitata del solito. Ciò si ottiene alterando il contenuto dei puntatori 55 e 56 (denominati, appunto, Top of Memory). Per rendervene conto, accendete il Commodore 64 e digitate:

PRINT FRE (0)

oppure, se il valore che ne risulta è negativo, battete PRINT 2↑16 +FRE(0)

Come risposta dovreste ottenere, se avete appena acceso il computer, il valore 38909 che corrisponde, grossomodo, al valore del messaggio che compare quando accendete un Commodore 64.

Se inoltre, digitate:

PRINT PEEK (55); PEEK (56)

ottenete come risposta: 0 e 160. Infatti 160x256 +0=40960

Tale valore, dovreste saperlo, corrisponde alla prima locazione di memoria che non risulta gestibile direttamente dal Basic. Da 49152 a 53247 è presente un'area di memoria RAM ("inaccessibile" al Basic) che può essere utilizzata per usi particolari, come routine in L.M. e utility di vario genere.

Chi si serve del Turbo Tape, come di tool di vario tipo (o delle routine grafiche di Toma), sa benissimo che quest'area risulta, purtroppo, inutilizzabile dato che è già occupata, appunto, da utility varie. Per continuare ad usare le utility cui il lettore è abituato e, contemporaneamente, quelle pubblicate nella rubrica Nuovo Sistema, ricorriamo ad un trucco semplicissimo che, nonostante riduca l'effettiva area a disposizione del Basic, lascia comunque una quantità di memoria ragguardevole ed evita il ricorso a tecniche (selezione del banco di memoria, utilizzo della RAM posta "sotto" le ROM eccetera).

Provate, infatti, a digitare:

POKE,80:CLR:PRINTFRE(0)

Otterrete in risposta: 18429 Bytes Free. Ciò vuol dire che, con la semplice operazione di una POKE, è possibile selezionare la quantità di memoria RAM gestibile da Basic. Sarà quindi possibile, nel caso specifico appena esaminato, utilizzare per gli scopi più disparati i (quasi) ventimila byte precedenti la locazione 40959. L'unica limitazione sarà costituita dalla lunghezza massima di un programma Basic che non deve superare i 18429 byte compresi quelli relativi a variabili, stringhe, matrici eccetera.

Utilizziamo "Top Memo"

Caricate il programma "Top Memo" pubblicato in queste pagine e, dopo il RUN, digitate 20000 come risposta alla domanda "Ultima locazione?".

Premuto il tasto Return appare il messaggio:

Ultima Locazione: 20000
Poke56,78:Poke55,33:
Clr:Print Fre(0)
Confermi? (S/N)

Se non siete soddisfatti del valore impostato, premete un tasto diverso da "S" ed il programma ripartirà.

In caso contrario, premendo "S", una tecnica particolare (incorporata nel programma), detta del Buffer di Tastiera, si incarica di alterare i puntatori 55 e 56 e di visualizzare la nuova quantità di memoria disponibile (circa 17657 byte). La reale quantità di memoria disponibile è, tuttavia, quella che rimane, dopo aver cancellato il programma Top Memo. Provate infatti (dopo aver registrato a parte il programma) a digitare:

NEW:PRINT FRE(0)

Otterete, in risposta, 17950.

Se avete seguito i nostri suggerimenti, nel vostro Commodore 64 sarà quindi presente una quantità di memoria RAM completamente indipendente dal Basic. Tale zona di RAM si estende dalla locazione 20000 a 40959 e sarà utilizzata, volta per volta, per allocare le varie routine in linguaggio macchina che costituiscono il vostro "NUOVO SISTEMA" personale.

```
100 PRINTCHR$(147)"FISSA TOP DI
     MEMORIA"
110 INPUT "ULTIMA LOCAZIONE";X:
    X=X+1
115 PRINTCHR$(147)"ULTIMA LOCAZ
    IONE:"X-1:GOSUB 180
120 PRINT:PRINT"CONFERMI? (S/N)
    "
130 GET A$:IF A$="" THEN 130
140 IF A$<>"S" THEN  RUN
150 PRINTCHR$(147);:GOSUB 180
160 POKE 198,3:POKE 631,19:POKE
     632,13:POKE 635,0
170 END
180 X1=INT(X/256):X2=X-(X1*256)
190 PRINT"P┌56,"X1":P┌55,"X2":C
    LR:?FRE(0)":RETURN
```

Come utilizzare il programma "Caricatore"

Dopo aver fissato il Top di memoria, e dopo aver eventualmente caricato la vecchia versione del NUOVO SISTEMA che si desidera aggiornare (seguita da NEW per "sistemare" i puntatori), caricate il "Caricatore".

Subito dopo digitate la routine che vi interessa che, come tutte le altre, inizia da riga 1000. Per sicurezza registrate e verificate, alla fine della trascrizione, il programma Basic così realizzato: sarà utilissimo in caso di errore.

A questo punto digitate il fatidico RUN e, alla domanda relativa all'indirizzo da cui volete allocare, digitate il valore scelto.

Dopo alcuni istanti verrà visualizzata la locazione iniziale e quella finale, occupate dalla routine, ed una richiesta di conferma. Digitando il tasto diverso da "S" il programma riparte.

Se tutto ci sembra a posto, invece, la pressione del tasto "S" visualizzerà nuovamente le istruzioni per il corretto uso della routine ed una serie di POKE (abbreviate) che, se vi posizionate sopra col cursore e premete il tasto Return, altereranno i puntatori in modo da registrare, su disco o nastro, solo la zona RAM appena allocata e che contiene la nostra routine LM.

Se, invece, volete allocare altre routine LM, seguite la procedura descritta per ciascuna routine avendo l'unica accortezza di non "sovrapporvi" ad altre routine già presenti in memoria. Solo alla fine potete registrare l'intero blocco di routine in un sol "colpo", ricorrendo al programma "Save Zona RAM".

```
100 REM PROGRAMMA CARICATORE
110 REM PER ENCICLOPEDIA
120 REM DI ROUTINE IN
130 REM LINGUAGGIO MACCHINA
140 :
150 REM PER UTILIZZARLO, LEGGI
    LE ISTRUZIONI PUBBLICATE SU
160 REM COMMODORE COMPUTER CLUB
170 :
180 Y=-1:GOSUB 1000:PRINT:INPUT
     "DA QUALE LOCAZIONE";X
190 READ W:Y=Y+1:IF W<0 THEN 21
    0
200 GOTO 190
210 PRINT"PRIMA LOCAZIONE ="X
220 PRINT"ULTIMA LOCAZIONE ="X+
    Y-1:PRINT
230 PRINT"CONFERMI? (S/N)"
240 GET A$:IF A$="" THEN 240
250 IF A$="S" THEN RESTORE :T=X
    :GOTO 280
260  RUN
270 :
280 GOSUB 1000:PRINT:PRINT"ATTE
    NDERE...":PRINT:W=0
290 READ B:IF B>=0 THEN POKE T,
    B:T=T+1:W=W+B:GOTO 290
300 READ B:IF B<>W THEN PRINT:P
    RINTCHR$(18)"ERRORE DI TRAS
    CRIZIONE":END
310 PRINT"ROUTINE ALLOCATA DA"X
    "A"T-1"COMPR.":PRINT
320 PRINT"ATTIVARE CON SYS"X:PR
    INT
330 X1=INT(X/256):X2=X-(X1*256)
    :PRINT"POKE44,"X1":POKE43,"
    X2;
340 Y1=INT(T/256):Y2=T-(Y1*256)
    :PRINT":POKE46,"Y1":POKE45,
    "Y2:END
```

Come utilizzare il programma "Save Zona RAM"

Dopo aver allocato le routine che più vi interessano (in aggiunta, magari, ad altre caricate precedentemente), caricate questo programma e lanciatelo col solito RUN.

Alla domanda "Prima locazione?" rispondete indicando il numero del byte che rappresenta il primo codice macchina della prima routine presente in memoria. Analogamente, alla domanda "Ultima locazione?" digitate il valore dell'ultimo byte dell'ultima routine LM presente.

Alla richiesta di conferma premete il tasto "S", indicate con 1 (disco) oppure 2 (nastro) il tipo di memoria di massa che intendete utilizzare e, in seguito, il nome che intendete assegnare al file.

Come già detto in altre parte dell'articolo, noi lo indicheremo con NUOVO SIS/LM ma voi, ovviamente, potrete assegnare il nome che più vi aggrada.

A questo punto il file—programma appena registrato conterrà tutte le routine LM di cui conserverete gelosamente la mappa della memoria allo scopo di utilizzarle in seguito. Ricordiamo che ogni routine viene eseguita dal comando SYS(XX) in cui "XX" è il valore del primo byte della routine in questione. E' ovvio che, in alcuni casi, è necessario settare altre locazioni prima del comando SYS. Ogni routine pubblicata, ovviamente, indicherà tutte le informazioni necessarie per un corretto utilizzo del vostro Nuovo Sistema.

```
100 REM PROGRAMMA IDONEO PER
110 REM REGISTRARE ZONE DI
120 REM MEMORIA AUTOMATICAMENTE
130 REM SU CASSETTA O DISCO
140 :
150 PRINTCHR$(147):INPUT "LOCAZ
    IONE INIZIALE";X
160 INPUT "LOCAZIONE FINALE";T:
    T=T+1
170 PRINT:PRINT"INIZIO:"X:PRINT
    "FINE:"T-1
180 IF T<=X THEN  RUN
190 PRINT:PRINT"CONFERMI? (S/N)
    "
200 GET A$:IF A$="" THEN 200
210 IF A$<>"S" THEN  RUN
220 PRINT:INPUT "NOME FILE";A$
230 PRINT"1- CASSETTA":PRINT"2-
    DISCO"
240 GET B$:IF B$="" THEN 240
250 IF B$="1" THEN W=1:GOTO 280
260 IF B$="2" THEN W=8:GOTO 280
270 GOTO 240
280 PRINTCHR$(147);
290 POKE 198,5:POKE 631,19:POKE
     632,13:POKE 633,13:POKE 63
    4,13:POKE 635,0
300 X1=INT(X/256):X2=X-(X1*256)
    :PRINT"P┌44,"X1":P┌43,"X2;
310 Y1=INT(T/256):Y2=T-(Y1*256)
    :PRINT":P┌46,"Y1":P┌45,"Y2
320 PRINT:PRINT:PRINT"S♠"CHR$(3
    4)A$CHR$(34)","W",1"
330 PRINT:PRINT:PRINT:PRINT:PRI
    NT"SYS64738"
```

GOTO calcolato (per C-64) (20000/20011)

di Giancarlo Mariani

Molti di voi, sviluppando programmi lunghi e complessi, avranno avuto difficoltà nell'uso dei salti condizionati. I problemi del tipo: "Come inserire in un programma molte linee del tipo IF... THEN GOTO..."? capitano quando si ha un numero elevato di controlli, all'interno di un programma Basic, che impongono la presenza di numerose linee del tipo esaminato. Ciò non solo peggiora le leggibilità del programma, ma lo allunga considerevolmente, creando problemi per ciò che riguarda l'occupazione di memoria.

Sarebbe molto più semplice e funzionale raggruppare tutte le linee in una sola per mezzo di una speciale istruzione del tipo CGOTO (GOTO Calcolato) che salta ad un determinato numero di linea elaborato in precedenza. Un'istruzione di questo tipo, come ben sappiamo, non è purtroppo presente nel Basic del C-64.

Per nostra fortuna i progettisti della Commodore hanno sviluppato un Sistema Operativo modulare ben strutturato, tale da permettere, utilizzando alcune routine dello stesso S.O , di "costruire" la CGOTO in soli 12 byte di LM.

Innanzitutto spieghiamo che cosa è e come funziona la routine proposta.

Questa permette di inserire, in un qualsiasi programma Basic, istruzioni simili a CGOTO A, oppure CGOTO 100+R*4, ed altre che, come intuibile, effettuano un salto alla linea calcolata dall'espressione (oppure alla variabile) presente dopo l'istruzione stessa. L'utilità di una istruzione del genere trova applicazione in una miriade di casi: si potrebbe introdurre direttamente, da tastiera, la linea da eseguire: oppure si potrebbe calcolare una formula e saltare in base al risultato di tale elaborazione eliminando, come detto prima, lunghe e noiose righe di IF... THEN... GOTO.

Simulare una CGOTO è estremamente semplice, ma, prima di utilizzarla, può essere interessante conoscere come funziona la GOTO "normale" presente nel C-64.

Dapprima (ed è proprio questo il punto che interessa maggiormente) la GOTO "prende" il numero presente dopo il comando stesso e, in seguito, controlla che tale valore rientri nei limiti ammessi (numero intero compreso tra 0 e 63999); quindi ricerca il numero di linea "incriminato": se questo non esiste all'interno del programma Basic, visualizza il messaggio "Undef'd statement error", altrimenti setta i puntatori del programma al nuovo numero di linea ed esegue quest'ultima.

Purtroppo, per un errore di programmazione del S.O. (non sappiamo se voluto o meno), la routine LM che "chiama" la GOTO, e che serve per individuare il numero di linea, accetta in ingresso solo valori numerici scritti in forma esplicita e non sotto forma di variabili o di espressioni numeriche.

Questa limitazione può facilmente essere aggirata: è risaputo che nel S.O. è presente anche la routine per accettare valori da calcolare mediante espressioni oppure variabili. Tale intuizione è confermata da una forma sintattica dell'istruzione POKE (tanto per citarne una), che, appunto, accetta per parametri anche variabili o espressioni numeriche.

Basterà quindi, per caricare il numero di linea da calcolare, utilizzare la stessa routine della POKE e saltare, in segui-

to, al "normale" GOTO, per ottenere lo scopo prefissato.

Il salto, a dire il vero, non sarà effettuato proprio all'inizio della routine GOTO, ma più "avanti" in modo da evitare l'operazione che esamina del numero di linea, dato che a ciò provvede la nostra routine LM.

Per realizzare il tutto è sufficiente effettuare solo quattro salti al sistema operativo, per un totale di appena dodici byte.

Chi desidera studiare il disassemblato, noterà che la prima istruzione (JSR$AEFD), serve per controllare la presenza di una virgola. Come, infatti, diremo in seguito, la sintassi della nuova istruzione sarà:
SYS XXXX,Espressione numerica
e se il carattere virgola (,) non risulta presente, viene visualizzato il solito SYNTAX ERROR.

I due JSR seguenti (JSR$AD8A e JSR$B7F7) sono gli stessi utilizzati dalla POKE Commodore (e da altre istruzioni analoghe) e servono, appunto, per caricare il numero, la variabile oppure una qualsiasi espressione (purchè numerica), presente dopo il comando GOTO. Se l'espressione non è numerica viene segnalato un TYPE MISMATCH ERROR.

Il quarto salto alla subroutine (JSR$A8A3) scavalca i tre byte presenti dopo la normale istruzione GOTO, evitando così di far eseguire alla stessa un lavoro già fatto in precedenza.

```
1000 PRINTCHR$(147)"QUESTA ROUTI
     NE CONSENTE DI EFFETTUARE U
     N GOTO CALCOLATO"
1010 RETURN
1020 DATA 32,253,174,32,138,173,
     32,247,183,76,163,168
1030 DATA -1,1671
```

Come utilizzare GOSUB calcolato

- *Caricare (o digitare dalla rivista) il programma caricatore dopo aver fissato il Top di memoria (mediante il programma Top Memo) a 20000.*
- *Digitare la routine GOSUB Calcolato pubblicata in queste pagine (numerata da 1000 in poi).*
- *Registrare e verificare il programma così realizzato.*
- *Digitare RUN ed indicare il valore 20012 (o altro che il lettore ritenga idoneo) alla domanda: "Da quale locazione?"*
- *Premere il tasto "S" alla domanda "Confermi?".*
- *A questo punto la routine è posizionata dalla locazione 20012 alla 20049 e si può attivare con SYS 20012.*
- *Il programma dimostrativo "Demo Gosub Calcolato", da caricare dopo aver seguito le fasi precedenti, costituisce un esempio di utilizzo.*

```
100 REM DEMO
110 REM GOTO CALCOLATO
120 INPUT "GOTO (140/150/160/17
    0/180)";X
130 SYS20000,X
140 PRINT"QUESTA E' LA RIGA 140
    ":GOTO 120
150 PRINT"QUESTA E' LA RIGA 150
    ":GOTO 120
160 PRINT"QUESTA E' LA RIGA 160
    ":GOTO 120
170 PRINT"QUESTA E' LA RIGA 170
    ":GOTO 120
180 PRINT"QUESTA E' LA RIGA 180
    ":GOTO 120
```

GOSUB calcolato (per C-64) (20012/20049)

di Giancarlo Mariani

Per la CGOSUB non c'è molto in più da dire di quanto non sia stato detto sulla CGOTO. Studiando il disassemblato del S.O., si nota che, come prima operazione, vengono registrati nello stack gli indirizzi di ritorno. Solo in seguito viene effettuato un salto all'istruzione GOTO.

Quando si incontrerà l'istruzione RETURN, verranno prelevati dallo stack gli indirizzi memorizzati nella fase precedente e ripristinato il normale flusso del programma.

Purtroppo la routine di GOTO non viene "chiamata" all'inizio del GOSUB, ma solo dopo aver salvato i vari puntatori, per cui è stato necessario copiare in RAM tutta la routine di GOSUB ed attuare su di essa le modifiche necessarie.

Chi ne ha voglia, può studiare il disassemblato: la routine, fino al JSR$0079, è identica a quella del S.O. (posizionata normalmente a $A883). In seguito il "solito" JSR$AEFD controlla la presenza della virgola ed i due JSR$AD8A e JSR$B7F7 "prendono" il numero di linea.

Si salterà quindi alla routine di GOTO+3 byte (JSR$A8A3) per le ragioni spiegate in precedenza.

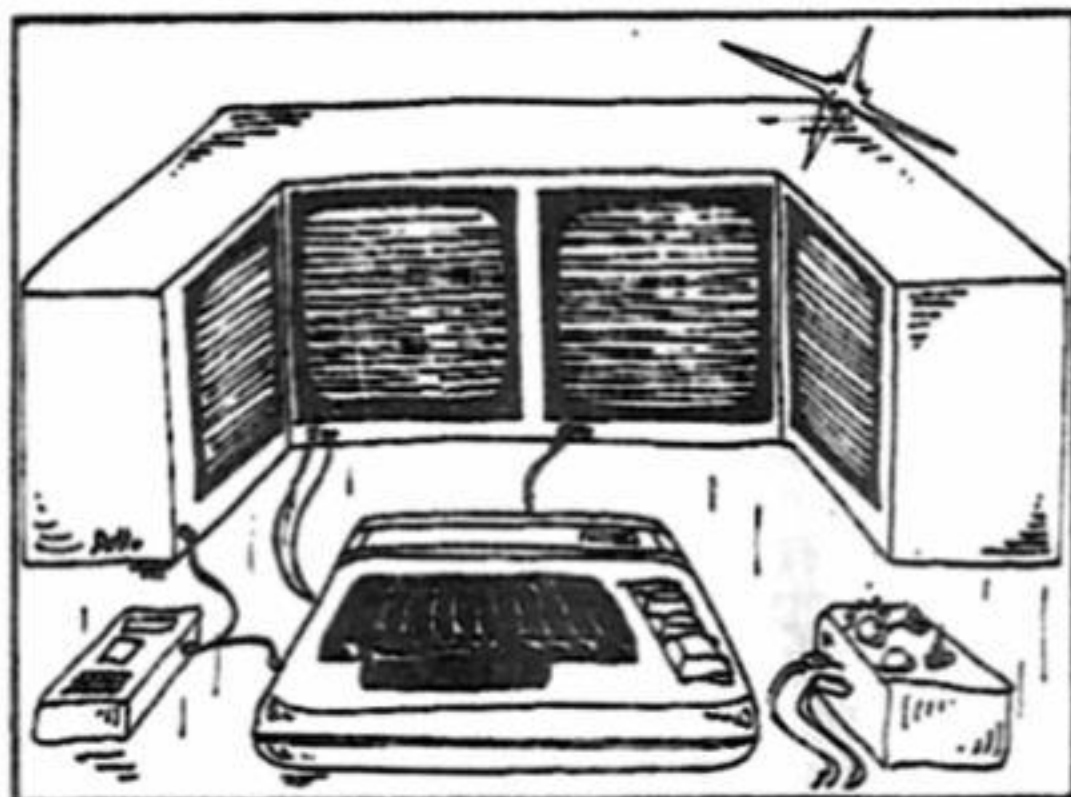
Dopo una CGOSUB si può inserire una qualsiasi altra istruzione sulla stessa linea Basic purchè, ovviamente, separata dal carattere di doppio punto (:). Questa sarà eseguita al ritorno dalla subroutine chiamata. A questo scopo serve l'ultimo salto (JMP$A7AE).

```
1000 PRINTCHR$(147)"QUESTA ROUTI
     NE CONSENTE DI EFFETTUARE U
     N GOSUB CALCOLATO"
1010 RETURN
1020 DATA 169,3,32,251,163,165,1
```

```
     23,72,165,122,72,165,58,72
1030 DATA 165,57,72,169,141,72,3
     2,121,0,32,253,174,32,138
1040 DATA 173,32,247,183,32,163,
     168,76,174,167
1050 DATA -1,4505
```

Come utilizzare GOTO calcolato

- *Caricare (o digitare dalla rivista) il programma caricatore dopo aver fissato il Top di memoria (mediante il programma Top Memo) a 20000.*
- *Digitare la routine "GOTO Calcolato" pubblicato in queste pagine (numerata da 1000 in poi).*
- *Registrare e verificare il programma così realizzato.*
- *Digitare RUN ed indicare il valore 20000 (o altro che il lettore ritenga idoneo) alla domanda: "Da quale locazione?".*
- *Premere il tasto "S" alla domanda "Confermi?".*
- *A questo punto la routine è posizionata dalla locazione 20000 alla 20011 e si può attivare con SYS 20000.*
- *Il programma dimostrativo "Demo Goto Calcolato", da caricare dopo aver seguito le fasi precedenti, costituisce un esempio di utilizzo.*

```
100 REM DEMO
110 REM GOSUB CALCOLATO
120 INPUT "GOSUB (140/150/160/1
    70)";X
130 SYS20012,X:GOTO 120
140 PRINT"QUESTA E' LA RIGA 140
    ":RETURN
150 PRINT"QUESTA E' LA RIGA 150
    ":RETURN
160 PRINT"QUESTA E' LA RIGA 160
    ":RETURN
170 PRINT"QUESTA E' LA RIGA 170
    ":RETURN
```

INTERP Stringa (per C-64) (20050/20128)

di Giancarlo Mariani

Vi è mai capitato di sviluppare un programma in grado di eseguire calcoli oppure di tracciare il grafico di funzioni matematiche, trigonometriche, o altro?

Il problema più difficile da risolvere, in casi come questi, risiede nel dover inserire, in qualche modo, la funzione da calcolare all'interno del programma.

Come ben sappiamo, non è possibile, ad esempio, assegnare: A$="Y=FUNZ(X)" e, in seguito, calcolare il valore di "Y" a seconda della funzione contenuta in A$.

Per aggirare l'ostacolo vengono, di norma, escogitati diversi metodi, uno dei quali consiste nel far scrivere sul video un numero di linea Basic, contenente la funzione da calcolare, e in seguito, ricorrendo al buffer di tastiera, simulare la pressione del tasto Return ottenendo, in tal modo, l'inserimento della nuova linea all'interno del programma.

Questo metodo funziona benissimo, solo che, per scrivere la linea Basic desiderata, è necessario cancellare lo schermo.

Al momento dell'inserimento, inoltre, tutte le variabili fino a quel momento elaborate vengono annullate con gli inconvenienti che è facile immaginare.

Per fortuna Mamma Commodore ci viene incontro (senza saperlo...), allocando nel S.O. alcune routine adatte a risolvere il problema in modo veloce e sicuro.

Esaminiamo passo per passo la soluzione che proponiamo.

Come detto prima, abbiamo bisogno di una routine che prenda in considerazione una stringa qualsiasi e riesca a tramutarla in espressione eseguibile da Basic.

La cosa è molto più semplice di quanto sembri. Esiste, nel C-64, una particolare zona di memoria chiamata Basic Input Buffer (posizionata da $0200) in cui vengono temporaneamente inseriti i comandi Basic prima di essere tokenizzati, ossia convertiti in codici di un solo byte.

Questi comandi vengono tokenizzati (appena si preme il tasto Return) ed eseguiti subito, se in modo diretto, oppure allocati in memoria, se preceduti da un numero di linea.

Da questo fatto si intuisce che, all'interno del S.O. deve esistere una routine che "prende" la stringa comando da $0200; deve esistere, inoltre, un'altra routine che la tokenizza e, infine, una terza routine che provvede ad eseguire il comando tokenizzato.

Basterà quindi trovare il modo di inserire la nostra stringa comandi a partire da $0200 e richiamare le routine interessate per eseguirla immediatamente.

Questo è proprio quello che fanno gli ottanta bytes LM del programma proposto.

Studiamo il disassemblato: per prima cosa viene controllato se si è in modo programma, in caso contrario è stampato

il messaggio ILLEGAL DIRECT. In seguito, dopo aver controllato la virgola di separazione (verrà spiegata dopo la sintassi del comando) si preleva la stringa presente dopo il comando stesso e la si trasferisce, tramite una breve routine, nel Basic Input Buffer.

Non ci si deve dimenticare a questo punto di "salvare" i puntatori del Basic, allo scopo di permettere una corretta ripresa dell'elaborazione dopo l'esecuzione del comando.

Dopo aver inserito uno zero al termine della stringa, per indicare la fine del comando, si predispone il modo diretto (ricordiamo che si può tokenizzare solo in modo diretto). In seguito vengono settati i puntatori del Basic ($7A e $7B) al comando da eseguire e, con JSR$A57C, si tokenizza il comando. Subito dopo, per mezzo di JSR$0073, si preleva il comando tokenizzato e lo si esegue con la routine posta in $A7ED. Terminata l'esecuzione, si ripristinano i puntatori del Basic e il modo programma, in modo da riprendere l'elaborazione esattamente dal punto in cui era stata sospesa.

La stringa rappresentante i comandi o le istruzioni da eseguire, può contenere qualsiasi comando Basic esclusi Goto e Gosub.

Un'altra limitazione è costituita dal fatto che è possibile inserire un solo comando alla volta. Non è quindi possibile assegnare più comandi separati dal doppio punto (:). Se un comando non viene riconosciuto sarà segnalato il solito SYNTAX ERROR.

```
1000 PRINTCHR$(147)"QUESTA ROUTI
     NE CONSENTE DI INTERPRETARE
      IL CONTENUTO DI ";
1010 PRINT"UNA STRINGA ALFANUMER
     ICA COME SE FOSSE UN GRUPPO
      DI COMANDI BASIC"
1020 RETURN
1030 DATA 165,157,240,5,162,21,1
     08,0,3,32,253,174,32,158
1040 DATA 173,32,130,183,165,122
     ,133,251,165,123,133,252,13
     2,2
1050 DATA 160,0,177,34,153,16,2,
     200,196,2,208,246,169,0
1060 DATA 153,16,2,169,128,133,1
     57,169,16,133,122,169,2,133
1070 DATA 123,32,124,165,32,115,
     0,32,237,167,169,0,133,157
1080 DATA 165,251,133,122,165,25
     2,133,123,96
1090 DATA -1,9497
```

Come utilizzare INTERP A$

- *Caricare (o digitare dalla rivista) il programma caricatore dopo aver fissato il Top di memoria (mediante il programma Top Memo) a 20000.*
- *Digitare la routine INTERP A$ pubblicata in queste pagine (numerata da 1000 in poi).*
- *Registrare e verificare il programma così realizzato.*
- *Digitare RUN ed indicare il valore 20050 (o altro che il lettore ritenga idoneo) alla domanda: "Da quale locazione?".*
- *Premere il tasto "S" alla domanda "Confermi?".*
- *A questo punto la routine è posizionata dalla locazione 20050 alla 20128 e si può attivare con SYS 20050.*
- *Il programma dimostrativo "Demo Interp A$", da caricare dopo aver seguito le fasi precedenti, costituisce un esempio di utilizzo.*

```
100 REM DEMO
110 REM INTERP A$
120 A=20:B=40:C=60
130 A$="PRINTA*B+(C-B/A)"
140 SYS20050,A$
150 INPUT "DIGITA IL COMANDO";A
    $
160 SYS20050,A$
```

Cambia colore al carattere (per C-64) (20129/20188)

di Federico Canetta

Questa breve routine consente di cambiare istantaneamente il colore di tutti i caratteri, che hanno lo stesso codice video, presenti sullo schermo.

Dovreste sapere che digitando, ad esempio, POKE1024,1 comparirà, nella cella video in alto a sinistra, il carattere "A" maiuscolo. Al carattere "B" corrisponde il codice 2, e così via.

La routine LM che, come tutte le altre, si attiva con la SYS opportuna, richiede, per il suo corretto funzionamento, altri due accorgimenti. Nella locazione 168 è necessario digitare il codice del carattere video che si desidera coinvolgere; analogamente nella locazione 169 si digiterà il codice del colore che si intende assegnare al carattere in oggetto.

```
1000 PRINTCHR$(147)"QUESTA ROUTI
     NE CAMBIA IL COLORE AL CARA
     TTERE IL CUI ";
1010 PRINT"CODICE DI SCHERMO E'
     RIPORTATO NELLA LOCAZIONE 1
     68.";
1020 PRINT"NELLA LOCAZIONE 169 D
```

```
     EVE ESSER PRESENTE IL CODIC
     E DEL COLORE ";
1030 PRINT"DESIDERATO"
1035 RETURN
1050 DATA 169,7,133,167,169,231,
     133,166,160,0,177,166
1060 DATA 197,168,208,15,24,165,
     166,133,170,165,167,105
1065 DATA 212,133,171,165,169,14
     5,170,56,165,166,233,1
1070 DATA 133,166,165,167,233,0,
     133,167,165,167,201,4
1080 DATA 240,3,24,144,211,165,1
     66,201,0,208,247,96
1090 DATA -1,8553
```

Come utilizzare CAMBIA COLORE

- *Caricare (o digitare dalla rivista) il programma caricatore dopo aver fissato il Top di memoria (mediante il programma Top Memo) a 20000.*
- *Digitare la routine "Cambia Colore al Carattere" pubblicata in queste pagine (numerata da 1000 in poi).*
- *Registrare e verificare il programma così realizzato.*
- *Digitare RUN ed indicare il valore 20129 (o altro che il lettore ritenga idoneo) alla domanda: "Da quale locazione?".*
- *Premere il tasto "S" alla domanda "Confermi?".*
- *A questo punto la routine è posizionata dalla locazione 20129 alla 20188 e si può attivare con SYS 20129.*
- *Il programma dimostrativo "Demo Cambia Colore al carattere", da caricare dopo aver seguito le fasi precedenti, costituisce un esempio di utilizzo.*

```
100 REM DEMO
110 REM CAMBIA COLORE AL CARATT
    ERE
120 PRINTCHR$(147):FOR I=0 TO 2
    55:POKE 1024+I,I:NEXT
130 FOR I=1 TO 7:PRINT:NEXT
140 INPUT "CODICE CARATTERE (0-
    255)";X
150 INPUT "CODICE COLORE (0-255
    )";Y
160 POKE 168,X:POKE 169,Y:SYS20
    129
170 PRINTCHR$(145)CHR$(145)CHR$
    (145):GOTO 140
```

Scroll Carattere (per C-64) (20189/20245)

di Federico Canetta

Questa routine, che sarà apprezzata soprattutto dai creatori di videogame, consente di spostare in basso di un rigo tutti i caratteri presenti sul video che hanno lo stesso codice schermo. Prima di attivare la SYS è necessario trascrivere nella locazione 168, mediante una semplice POKE, il codice video desiderato.

```
1000 PRINTCHR$(147)"QUESTA ROUTI
     NE EFFETTUA UNO SCROLLING V
     ERSO IL BASSO DI"
1010 PRINT"TUTTI I CARATTERI AVE
     NTI LO STESSO CODICE DI SCH
     ERMO"
1020 PRINT"IL CODICE DEL CARATTE
     RE DA SCROLLARE"
1030 PRINT"DEVE ESSERE INSERITO
     NELLA LOCAZIONE 168"
1040 RETURN
1050 DATA 169,7,133,167,169,191,
     133,166,160,0,177,166
1060 DATA 197,168,208,12,160,0,1
     69,32,145,166,165,168
1070 DATA 160,40,145,166,56,165,
     166,233,1,133,166,165
1080 DATA 167,233,0,133,167,165,
     167,201,4,240,3,24
1090 DATA 144,214,165,166,201,0,
     208,247,96
1100 DATA -1,7769
```

Come utilizzare "Scroll Carattere"

- *Caricare (o digitare dalla rivista) il programma caricatore dopo aver fissato il Top di memoria (mediante il programma Top Memo) a 20000.*
- *Digitare la routine "Scroll Carattere" pubblicata in queste pagine (numerata da 1000 in poi).*
- *Registrare e verificare il programma così realizzato.*
- *Digitare RUN ed indicare il valore 20189 (o altro che il lettore ritenga idoneo) alla domanda: "Da quale locazione?".*
- *Premere il tasto "S" alla domanda "Confermi?".*
- *A questo punto la routine è posizionata dalla locazione 20189 alla 20245 e si può attivare con SYS 20189.*
- *Il programma dimostrativo "Demo Scroll Carattere", da caricare dopo aver seguito le fasi precedenti, costituisce un esempio di utilizzo.*

```
100 REM DEMO
110 REM SCROLL CARATTERE
120 PRINTCHR$(147):FOR I=1 TO 7
    :FOR J=1 TO 10:PRINT"A";:NE
    XTJ:PRINT:NEXTI
130 FOR I=1 TO 13:POKE 168,1:SY
    S20189:NEXT
```

Cancella Carattere (per C-64) (20246/20302)

di Federico Canetta

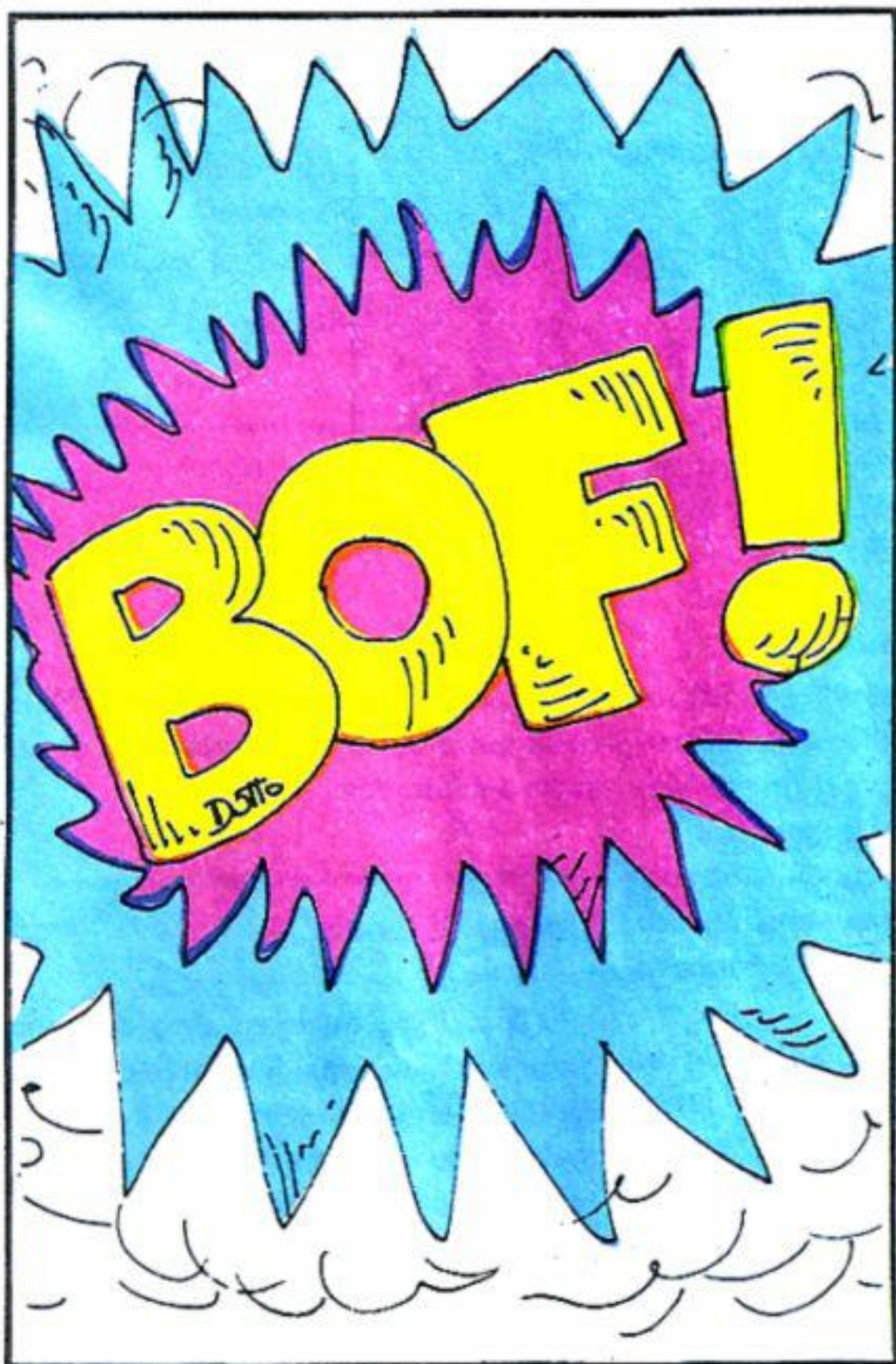
Questa routine, dal funzionamento analogo a quelle precedenti, consente di cancellare istantaneamente tutti i caratteri che, presenti sul video, hanno lo stesso cidice di schermo.

Prima di attivare la SYS è necessario trascrivere nella locazione 168 il codice video desiderato.

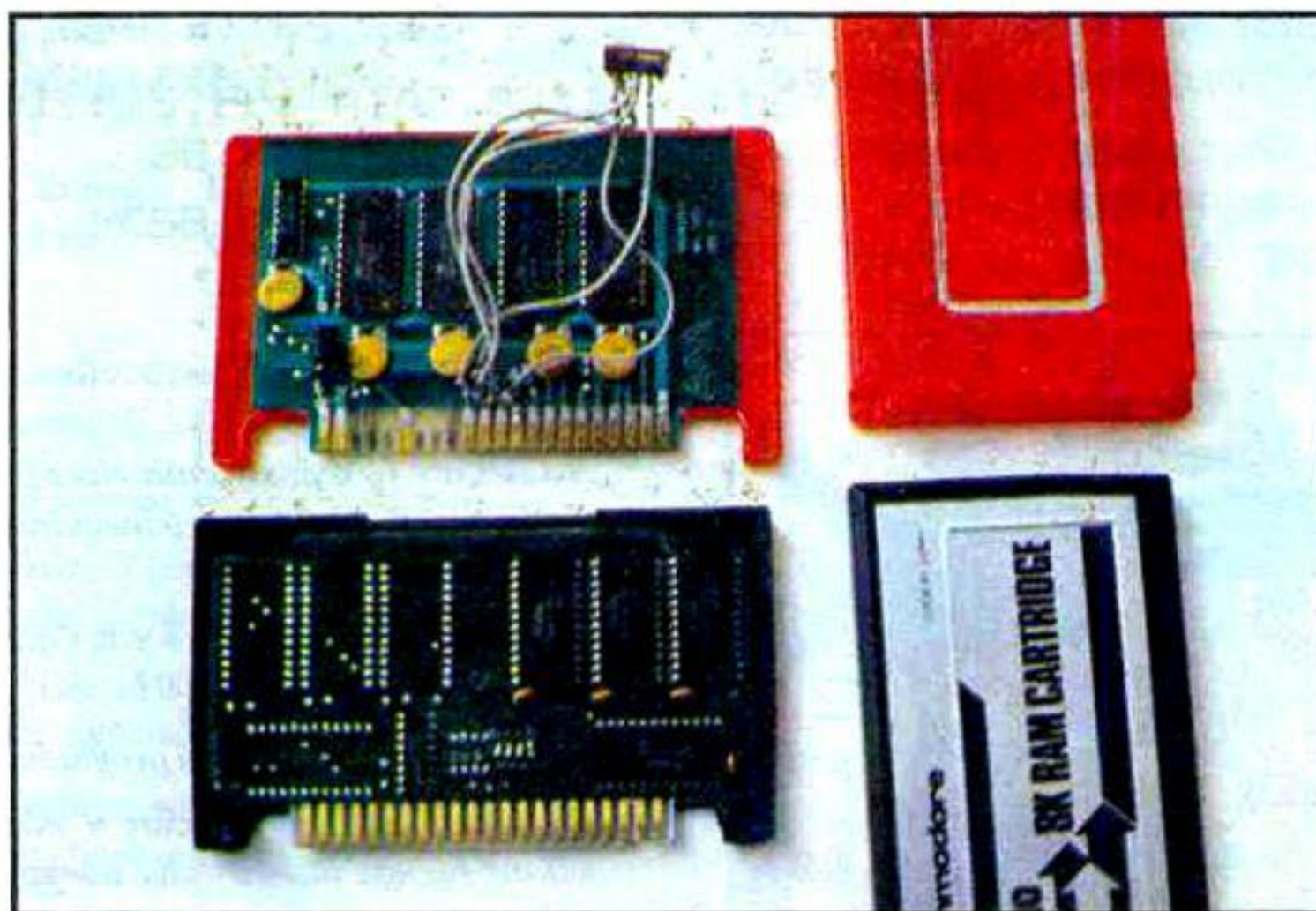
```
1000 PRINTCHR$(147)"QUESTA ROUTI
     NE CANCELLA DALLO SCHERMO I
      CARATTERI";
1010 PRINT" IL CUI CODICE (DI SC
     HERMO) VA INSERITO NELLA LO
     CAZIONE 168"
1020 RETURN
1030 DATA 169,7,133,167,169,231,
     133,166,160,0,177,166
1035 DATA 197,168,208,12,160,0,1
     69,32,145,166,165,168
1045 DATA 234,234,234,234,56,165
     ,166,233,1,133,166,165
1050 DATA 167,233,0,133,167,165,
     167,201,4,240,3,24
1060 DATA 144,214,165,166,201,0,
     208,247,96
1070 DATA -1,8234
```

Come utilizzare "Cancella Carattere"

- *Caricare (o digitare dalla rivista) il programma caricatore dopo aver fissato il Top di memoria (mediante il programma Top Memo) a 20000.*
- *Digitare la routine "Cancella Carattere" pubblicata in queste gine (numerata da 1000 in poi).*
- *Registrare e verificare il programma così realizzato.*
- *Digitare RUN ed indicare il valore 20246 (o altro che il lettore ritenga idoneo) alla domanda: "Da quale locazione?".*
- *Premere il tasto "S" alla domanda "Confermi?".*
- *A questo punto la routine è posizionata dalla locazione 20246 alla 20302 e si può attivare con SYS 20246.*
- *Il programma dimostrativo "Demo Cancella Carattere", da caricare dopo aver seguito le fasi precedenti, costituisce un esempio di utilizzo.*

```
100 REM DEMO
110 REM CANCELLA CARATTERE
120 PRINTCHR$(147):FOR J=65 TO
    69:FOR I=1 TO 10:PRINTCHR$(
    J);:NEXTI:PRINT:NEXTJ
130 PRINT:PRINT
140 INPUT "CARATTERE (1-5)";X
150 POKE 168,X:SYS20246:PRINTCH
    R$(145);:GOTO 140
```

Come ti sproteggo il cartridge

Qualche goccia di stagno, un'espansione da 8K Ram e un po' d'ingegno per diffondere uno dei segreti di Pulcinella.

di Alessandro de Simone

Sappiamo benissimo a che cosa state pensando: dopo mesi e mesi di discorsi paternalistici contro le persone cattive che sproteggono il software delle persone buone, ecco che quelli di CCC si apprestano a descrivere in dettaglio una tecnica di sprotezione.

Niente paura, amici! Anche noi, come ogni malfattore che si rispetti, abbiamo una scusa pronta per giustificarci, anzi quattro:

- *Il Vic 20 è ormai uscito di produzione e, di conseguenza, non danneggiamo nessuno.*
- *La tecnica che descriviamo è valida solo per chi possiede il drive e richiede interventi hardware sul computer (pochi hanno il coraggio di metterci le mani).*
- *La sprotezione descritta è un'ottima occasione per descrivere una nuova tecnica di protezione per coloro che fabbricano cartucce.*
- *In genere i cartridge per Vic 20 sono dei videogiochi che rendono stupidi gli utilizzatori e, sproteggendo, scoraggiamo la produzione di accessori nefandi per le giovani generazioni (così imparano).*

Ed ora, dopo aver contrabbandato per buoni, comportamenti che meriterebbero la sedia elettrica, consentiteci di istigarvi a delinquere.

Un po' di teoria

Il Vic 20, e qualsiasi altro computer domestico, dispone di un'area di memoria RAM e di una seconda area che, normalmente, è "vuota" ma che potrebbe ospitare una cartuccia piena di altra RAM (espansione di memoria) oppure ROM (cartucce giochi, interfacce, diavolerie varie).

Nel caso specifico del Vic 20 le locazioni destinate alla cartuccia sono quelle numerate da 40960 a 49152.

Nonostante debba esser noto, ribadiamo il fatto che un computer non può sapere se una memoria è RAM oppure ROM oppure EPROM o altro.

Se, quindi, in quel banco di 8K inseriamo una cartuccia (prima di accendere il Vic) il nostro sistema non avrà più un "vuoto" in quella particolare zona.

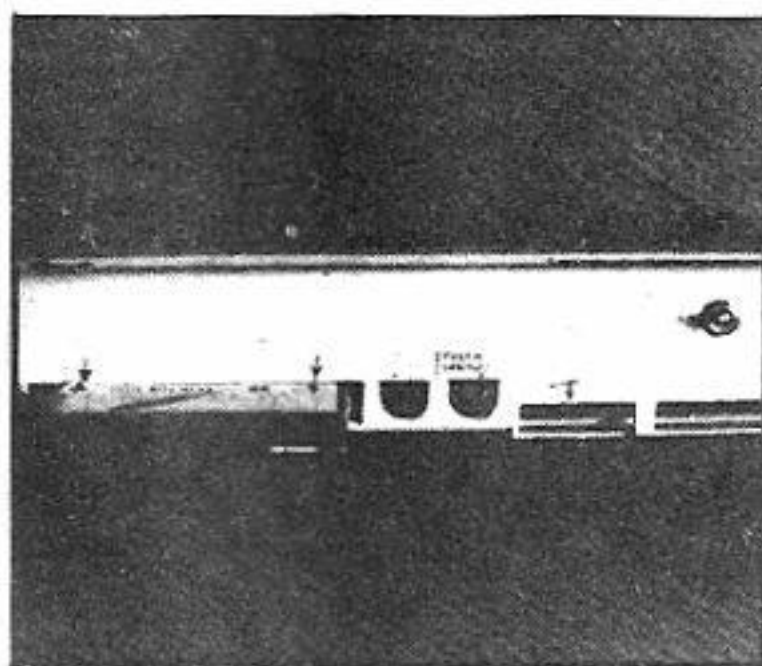
Dovreste anche sapere che i sistemi Commodore sono strutturati in modo tale che, al momento dell'accensione, se in alcune particolari locazioni di memoria sono presenti determinati codici, il sistema "riconosce" la presenza di una cartuccia e viene attivata una routine in linguaggio macchina che fa partire il programma.

E' ovvio che i costruttori di cartucce conoscono benissimo questo particolare e, in base ad esso strutturano le cartridge che producono.

Per chiarire quali sono i codici è bene riferirsi alla seguente tabella che contiene nella colonna di sinistra l'indirizzo e in quella di destra il valore del loro contenuto:

Indir.	Peek
40960	9
40961	160
40962	9
40963	160
40964	65
40965	48
40966	195
40967	194
40968	205

Di queste nove locazioni le ultime tre rappresentano i caratteri maiuscoli "C" "B" "M" che, come è noto, sono le iniziali di Commodore Business Machines. Le due precedenti sono un vero e proprio codice (65=a; 48=0).

Quando il Vic 20 viene acceso, il sistema operativo, prima di effettuare altri controlli, verifica se, nelle locazioni da 40964 a 40968, è presente il messaggio "a0CBM". In caso negativo continua per la sua strada fino alla visualizzazione del solito "Commodore... Basic V2... Bytes Free...".

Se, invece, i cinque caratteri sono riconosciuti, il computer deduce che è presente una cartuccia nell'apposito alloggiamento e, invece di continuare le solite operazioni, salta ad eseguire il programma in L.M. la cui locazione iniziale è data dai contenuti delle locazioni 40962 e 40963. Le prime due locazioni (40960 e 40961) hanno il compito di ricordare al computer il programma LM da attivare in caso di pressione di Run/Stop e Restore.

Nel caso che abbiamo indicato in tabella, una cartuccia che presentasse quei particolari valori partirebbe anche con SYS(160*256+9). E diciamo "partirebbe" perchè in effetti, non è possibile digitare SYS... dato che, non appena il computer viene acceso, il programma si lancia da solo e non consente, normalmente, il "recupero" del Basic.

Abbiamo però detto che tutto ciò avviene solo se, al momento dell'accensione, il computer individua il codice "a0CBM". Se, dunque, potessimo inserire la cartuccia DOPO aver acceso il calcolatore (a confronto, cioè, avvenuto e fallito), potremmo leggere ed esaminare comodamente il programma tramite Basic.

Come è noto, però, è pericoloso inserire una cartuccia a computer acceso e, anche riuscendoci, si generano malfunzionamenti di vario tipo.

Il trucco

Il rimedio all'"inconveniente" consiste, semplicemente, nello sconnettere il pin di alimentazione del connettore per cartucce e collegarlo ad un interruttore che troverà posto sul retro del Vic. Poichè questa operazione chirurgica è bene sia fatta solo da persone esperte, ci limiteremo a dire che il pin dei 5 volt è il N.21 del connettore: chi se ne intende non avrà difficoltà ad interpretare quanto abbiamo detto; gli altri, che non hanno capito niente, è bene che non ci mettano le mani. Dato che ci siete, inoltre, e stabilito che ormai il vostro Vic è manomesso, aggiungete anche un comodo pulsante di Reset incorporato: è il pin indicato con "X" nell'appendice del manuale di istruzione del computer.

Supponendo di aver realizzato le connessioni descritte, come si può fare per esaminare il contenuto di un cartridge?

- *Spegnete il computer.*
- *Inserite la cartuccia che intendete sproteggere.*
- *Posizionate l'interruttore dei 5 volt in modo "spento".*
- *Accendete il computer: compare il solito messaggio relativo al Vic 20 senza espansioni di memoria.*
- *Accendete l'interruttore dei 5 volt: non dovrebbe succedere assolutamente nulla (almeno ve lo auguriamo...)*
- *Da questo momento potete leggere, tramite Peek, il contenuto degli 8K Ram della cartuccia.*

Se tutto è in ordine il programma dovrebbe partire con SYS64802, oppure premendo il pulsante di Reset (se l'avete inserito) e ancora, più semplicemente, con Run/Stop e Restore. E' ovvio che alcune cartucce potrebbero avere modi di lancio differenti.

Come registrarlo

Da quanto detto, però, il programma su cartridge può essere esaminato ma, agli atti pratici, potrebbe non interessare

nessuno: la sprotezione, si sa, è bella se consente copie illegali. Che ce ne facciamo di un sistema che consente soltanto di STUDIARE (orrore!) un programma?

Tutti sanno (!) che è possibile registrare una qualsiasi zona di memoria alterando, opportunamente, i puntatori 43, 44 (inizio Basic) e 45, 46 (inizio variabili, fine implicita del Basic). Lo faremo anche noi; digitate:

POKE43,0: POKE44,160:
POKE45,255: POKE46,192

Il Vic 20 risponderà: Ready. Digitate allora:

SAVE"NOME",8,1

A questo punto il vostro drive ronzerà tenendovi col fiato sospeso per il tempo necessario a registrare 8K Byte.

Sul vostro disco, alla fine, è presente l'immagine del programma su cartuccia assolutamente inservibile...

Ma perchè ci dite ciò?

Avete letto benissimo: il file programma registrato non potrà essere utilizzato. A che cosa serve, allora?

Calma ragazzi, dateci il tempo di spiegare qualcosa prima di menare le mani.

Il programma registrato, come, del resto, quello sulla cartuccia "originale", è un programma in L.M. che, per funzionare, ha bisogno di esser fisicamente presente proprio nel banco di memoria posizionato da 40960 a 49151. Poichè l'espansione di memoria di 8K Ram occupa le locazioni da 8192 a 16383 il programma, caricato in questa zona, non potrà mai funzionare. Per farlo girare abbiamo bisogno di un'espansione di memoria da 8K Ram che occupi le stesse locazioni del programma copiato (40960 49151).

E' bene ricordare che le Ram sono tutte eguali e che la zona di memoria che occupano dipende da un pin che, collegato ad uno dei quattro blocchi disponibili, assegna automaticamente un banco di memoria.

Ciò significa che un blocco di 8K di Ram che abbia il pin (non diciamo quale) collegato col pin 10 della porta di espansione, configurerà automaticamente il sistema come espanso ad 8K: quello col pin 10 è infatti il collegamento di una comune espansione da 8K.

Ecco di seguito i pin della porta di espansione, il numero di blocco e le locazioni di memoria interessate.

La tabella dei pin

Pin	Blk	Locazioni	
10	1	8192	16383
11	2	16383	24575
12	3	24576	32767
13	5	40960	49151

Notiamo esplicitamente che il blocco 4 (da 32768 a 40959) è riservato al sistema e non può essere occupato da nessun "aggeggio".

I vari casi possibili

Se disponete di un'espansione di memoria di 8K originale Commodore, e la aprite, noterete con gioia che, al suo interno, sono presenti quattro micro interruttori (vedi foto) uno solo dei quali è acceso mentre gli altri tre sono spenti. Quello acceso è collegato, inutile dirlo, al pin 10 che configura automaticamente il sistema espanso di 8K all'accensione. Spegnendo tale interruttore, ed accendendo UNO SOLO degli altri tre, selezionerete, a seconda dei casi, uno dei quattro blocchi possibili.

Pertanto, per caricare il programma prima registrato su disco, dovete:

- *Aprire la cartuccia da 8K Ram Commodore.*
- *Accendere il primo interruttore e spegnere tutti gli altri.*
- *Richiudere la cartuccia ed inserirla nel Vic SPENTO.*
- *Accendere l'interruttore dei 5 volt (da voi installato prima).*
- *Accendere il computer e il drive.*
- *Digitare LOAD"NOME",8,1*
- *Al termine del caricamento digitare SYS 64802 oppure Run/*Stop e Restore oppure premere il pulsante di Reset: il gioco dovrebbe partire.

Paradossalmente, tenendo conto del costo maggiore, l'espansione da 16K Ram della Commodore non possiede microinterruttori al suo interno, ma solo saldature che presentano maggiori problemi.

Cartucce Ram di altre marche, analogamente, sono prive dei comodi interruttori che, praticando una fessura sulla cartuccia, sono facilmente gestibili senza esser costretti ad aprire e richiudere tutte le volte la custodia stessa. In questi frangenti i casi sono due: o vi arrangiate a dissaldare le piste inserendo interruttori, o lasciate perdere il progetto.

Come proteggere

Abbiamo detto, all'inizio, che avremmo suggerito un sistema per proteggere il proprio sofware da copiature simili a quella appena descritta.

Vediamo di riassumere brevemente come aggirare l'autorun:

• *Non appena il computer viene acceso il programma parte impedendo tentativi di salvataggio.*
• *Non è però possibile stabilire se il programma risiede su Rom (cartridge originale) oppure su Ram (cartridge sprotettore) e, in quest'ultimo caso, l'autorun non può funzionare.*

La soluzione

Il problema sarebbe bell'e risolto se il computer fosse così intelligente da sapere se il banco di memoria che ospita il programma è di tipo Rom (originale) oppure Ram (copia illecita).

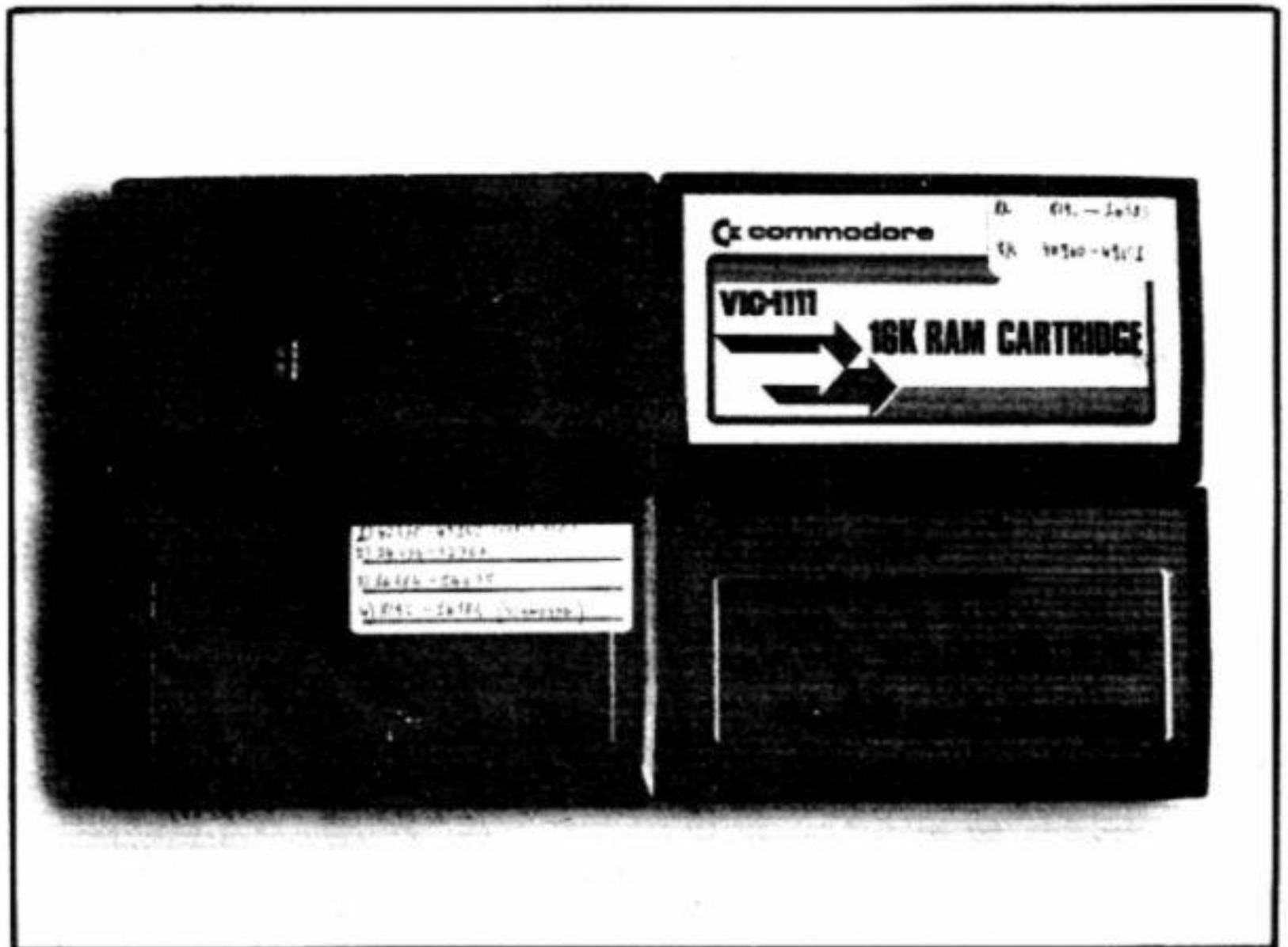
Il computer, forse, non è così intelligente, ma il programmatore lo è di certo.

Partiamo dunque dal presupposto che, benchè il banco di memoria sia lungo 8K, non tutti i suoi byte siano indispensabili al suo corretto funzionamento. In altre parole: se il vostro computer dispone di 3K non è indispensabile che un programma sia lungo 3000 byte per funzionare; può esser certamente più breve e, in questo caso, alcuni byte non saranno assolutamente utilizzati durante il suo funzionamento.

Supponiamo, tornando al nostro caso, che gli ultimi (o, addirittura, soltanto l'ultimo) byte non sia necessario per il funzionamento del programma. All'interno di questo, furbescamente, il softwarista inserirà un gruppo di istruzioni (da tradurre opportunamente in LM), che simbolicamente possiamo così indicare:

```
A=49151
POKE A,0
IF PEEK(A) THEN CONTINUE
POKE A,255
IF PEEK(A)< 255 THEN CONTINUE
RESET
```

Esaminiamole una per una:

• *Viene azzerata la locazione 49151 (l'ultima del banco).*
• *Se, ad una successiva, immediata lettura della stessa locazione il valore è diverso da zero (IF PEEK...) allora vuol dire che il precedente tentativo di scrittura (POKE A,0) è fallito. Se ne deduce che nella zona di memoria è presente una Rom (è impossibile, infatti, scrivere su una Rom, ma solo su una Ram). Questa, con molta probabilità, è quella originale ed il programma può quindi continuare.*
• *Se, invece, la lettura della locazione 49151 è eguale a zero, potrebbe significare che al suo posto è presente una Ram, azzerata dalla precedente operazione di Poke. Potrebbe, però, esserci al suo posto una Rom con valore zero.*
• *Per esser sicuri che in 49151 non ci sia un Rom nulla, cerchiamo di Pokarci 255.*
• *Se, nonostante Poke(A),255 la successiva lettura fornisce un valore diverso, vuol dire che è presente una Rom.*
• *In caso contrario, se siamo giunti fino alla fine, vuol dire che siamo in presenza di una Ram, ed il programma continua con un Reset che può esser di natura più varia possibile: un loop chiuso, disabilitazione permanente della tastiera, eccetera.*

Questo tipo di protezione è ormai costantemente adottato dai produttori di cartridge, specialmente per Commodore 64. I possessori di questo computer, infatti, potrebbero sproteggere nel modo più semplice possibile dato che il C/64 è pieno zeppo di Ram, anche nella zona normalmente occupata dal cartridge.

Se infatti tentate di sproteggere nel modo descritto una cartuccia per C/64, vi accorgete che non è così semplice come per il Vic 20.

E questo è anche uno dei motivi per cui non abbiamo avuto molti scrupoli a pubblicare il presente articolo...

Per i più bravi

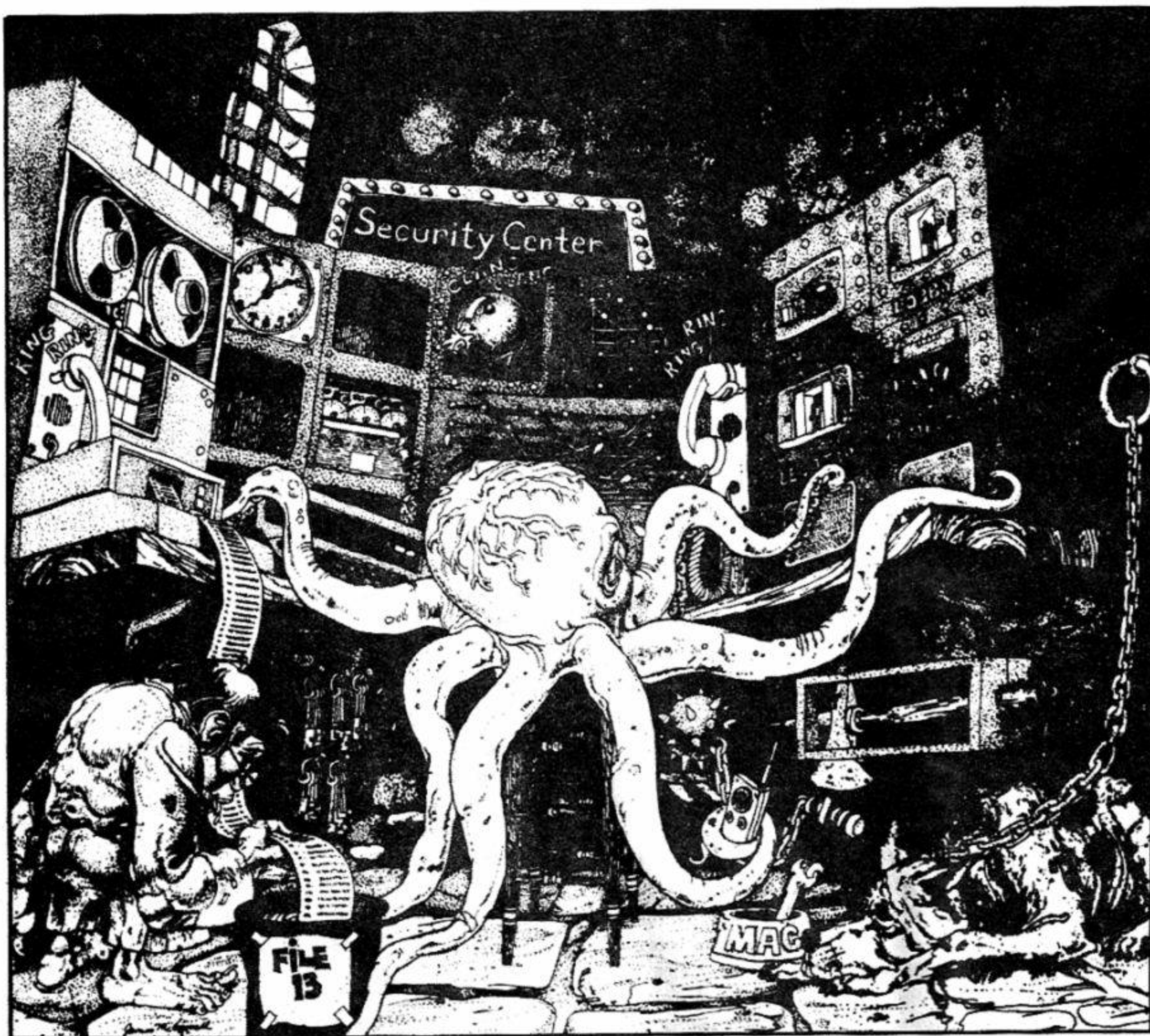
Tra le lettere dei lettori che descriveranno nel modo più semplice, chiaro e, soprattutto, BREVE, il perchè dell'alterazione del puntatore 45 (posto a 255 invece che a zero) insieme al sistema per trasferire su nastro una cartuccia, selezioneremo, come al solito, alcuni nominativi ai quali invieremo un appendiabiti ed un accendisigari.

I nominativi di coloro che riceveranno il chiodo ed il fiammifero saranno pubblicati per la gioia dei partecipanti.

P.S. L'importante non è vincere, ma partecipare.

Alla (ri)scoperta dell'Interrupt

Come costringere il cervello del computer ad eseguire di continuo una nostra routine

di Michele Maggi

Nonostante l'argomento sia stato già trattato più volte su C.C.C., lo affrontiamo nuovamente sia perché molti lettori, che da poco seguono la rivista, chiedono di parlarne sia perché una serie di note sull'interrupt sarà di certo utile anche all'utente più smaliziato.

La teoria

Non tutti sanno (ma dovrebbero saperlo) che anche quando il nostro computer sembra non far nulla, come ad esempio quando è stato appena acceso e vediamo il cursore lampeggiare, in realtà esegue milioni di operazioni senza che noi ce ne accorgiamo.

Ciò accade grazie alla notevole velocità di elaborazione del microprocessore che, ogni sessantesimo di secondo, interrompe il programma che sta eseguendo (posto che ne stia eseguendo uno) ed effettua una serie di controlli, necessari al corretto funzionamento del computer.

Tali controlli si riferiscono essenzialmente alla "scansione" della tastiera, per verificare se è stato premuto un tasto, al controllo del lampeggio del cursore, all'esame di un eventuale "dialogo" in corso tra computer e periferiche, all'aggiornamento della variabile del tempo "TI" e altri ancora.

Come inserirsi nell'Interrupt

Da quanto detto sembra di non poter far nulla per "fermare" o modificare la routine di controllo (detta appunto di interrupt) che, data la velocità con cui si svolge, da Basic rende impossibile un "inserimento" all'interno della sua struttura.

La richiesta di interrupt è infatti generata via hardware e, se ciò non bastasse, la stessa routine è situata in una zona di memoria a sola lettura (ROM).

Fortunatamente i progettisti del C64 hanno fatto in modo che l'indirizzo di partenza di questa routine fosse situato all'interno di due locazioni RAM i cui indirizzi esadecimali sono $0314 e $0315.

Grazie proprio a quest'ultima osservazione, nulla impedisce di modificare il contenuto delle due locazioni inserendo, al loro posto, l'indirizzo di una routine in LM da noi creata.

Agendo in tal modo, ogni volta che verrà generato un interrupt (e cioè 60 volte al secondo) il controllo passerà alla routine il cui indirizzo è stato modificato. Il risultato sarà un'esecuzione praticamente "continua" della nostra routine.

Un accorgimento da non dimenticare per nessun motivo è costituito dal fatto che, al termine della routine, sarà indispensabile comunque un salto alla originaria routine di Interrupt ($EA31).

Da quanto detto è agevole capire che la manipolazione del vettore di interrupt apre tantissime porte per ciò che riguarda effetti particolari che potranno aggiungere una nota di professionalità ai vostri programmi.

Due esempi pratici

In queste pagine sono pubblicati due esempi piuttosto banali, ma di sicuro "effetto", che si basano sulla tecnica descritta.

Lampeggio del bordo. Il bordo del video lampeggerà di continuo.

Scritta indelebile. Sulla prima riga dello schermo comparirà il messaggio "Commodore 64" e, qualunque tasto voi premiate (Shift e Clr/Home compreso), rimarrà sempre visibile.

Premendo i tasti Run/Stop e Restore si riportano i vettori alla normalità mentre, per far ripartire i programmi, bisognerà digitare SYS 828.

I più abili, armati di pazienza, possono disassemblare le routine proposte in modo da impadronirsi delle tecniche descritte, fra le più interessanti e stimolanti che è possibile realizzare.

```
10 REM ESEMPIO DI INTERRUPT
15 REM MESSAGGIO PERMANENTE
20 FOR I=828 TO 895:READ A:POK
   E I,A:X=X+A:NEXT
25 IF X<>4126 THEN PRINT"ERROR
   E DI DIGITAZIONE":END
30 SYS828
35 DATA  120,169,73,141,20,3,1
   69,3,141,21,3,88,96,162,40,
   189
40 DATA  87,3,157,255,3,202,20
   8,247,76,49,234,32,32,32,32
   ,32
45 DATA  32,32,32,32,32,32,32,
   32,3,15,13,13,15,4,15,18
50 DATA  5,32,32,54,52,32,32,3
   2,32,32,32,32,32,32,32,32
55 DATA  32,32,32,32
```

```
10 REM  ESEMPIO DI INTERRUPT
20 REM  BORDO LAMPEGGIANTE
30 REM  PER C-64
40 FOR I=828 TO 851:READ A:POK
   E I,A:X=X+A:NEXT
50 IF X<>2640 THEN PRINT"ERROR
   E DI DIGITAZIONE":END
60 SYS828
70 IF X<>2640 THEN PRINT"ERROR
   E DI DIGITAZIONE":END
80 SYS828
90 DATA 120,169,73,141,20,3,16
   9,3
92 DATA 141,21,3,88,96,162,0,1
   42
94 DATA 32,208,232,208,250,76,
   49,234
```

Enciclopedia di routine

di Maurizio Dell'Abate

12400 Print Using

(Per qualsiasi Commodore)

Sebbene sia già stata pubblicata una routine che compie le stesse funzioni (N.29 di C.C.C.) la riproponiamo, con la stessa numerazione, in modo che il lettore possa scegliere quella che ritiene più idonea.

"Print Using" vi permette di formattare un dato numerico, incolonnando il punto decimale.

Per "formattare" si intende stabilire il numero di cifre a sinistra e a destra del punto decimale da prendere in considerazione, in modo da rendere più gradevole l'Output su video (e/o su carta).

Prima di chiamare la subroutine è necessario assegnare alla variabile numerica X1 il valore da formattare ed alla variabile alfanumerica Y1$ il formato.

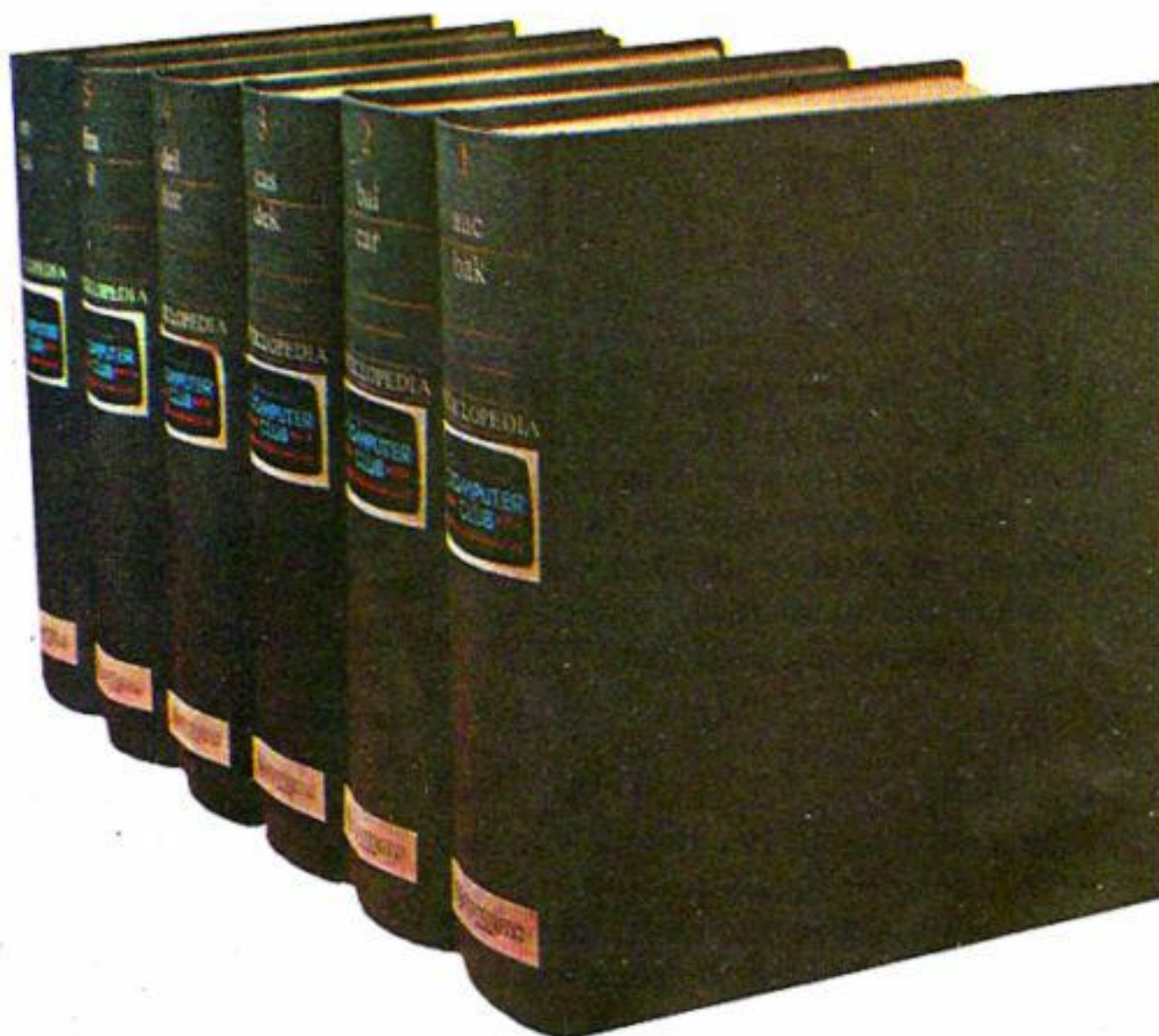
Esempio di formato:
Y1$="******.**"

La quantità di asterischi (potete usare al loro posto un qualsiasi altro carattere) presenti prima e dopo il punto, indica alla subroutine il numero delle cifre da considerare. Al ritorno la variabile stringa XY$ conterrà il valore opportunamente formattato e, se necessario, viene inserito un certo numero di spazi bianchi, in modo da consentire l'allineamento della virgola (punto decimale).

I seguenti esempi dovrebbero chiarire il funzionamento della routine; il carattere apostrofo (') è da intendersi come uno spazio bianco:
Y1$="**** ***"

X1=535.35367
Risultato:
XY$="'535.353"

X1=43437.65
Risultato:
XY$="3437.65"

X1=89
Risultato:
XY$="''89"

Il Demo pubblicato insieme con la subroutine è molto semplice e contiene una schermata di spiegazione.

```
100 REM ESEMPIO D'USO
110 REM USING
120 REM PER QUALSIASI COMMODORE
130 :
140 PRINTCHR$(147)
150 PRINT"SULLA SINISTRA DEL VI
    DEO APPARIRA'"
160 PRINT"UNA COLONNA DI NUMERI
     CASUALI, SULLA"
170 PRINT"DESTRA GLI STESSI NUM
    ERI, STAMPATI"
180 PRINT"PER MEZZO DELLA ROUTI
    NE PROPOSTA."
190 PRINT"(FORMATO: ****.***).
200 PRINT"GIUDICA TU QUALE OUTP
    UT E' PIU'"
210 PRINT"GRADEVOLE..."
220 PRINT"PREMI UN TASTO PER IN
    IZIARE "
230 GET A$:IF LEN(A$)=0 THEN 23
    0
240 PRINTCHR$(147):Y1$="****.**
    *":FOR I=1 TO 20
250 X1=INT(RND(1)*2000000)/(RND
    (0)*2000)
260 IF RND(1)<.2 THEN X1=X1+100
    00*RND(1)
265 IF RND(1)<.3 THEN X1=INT(X1
    )
267 IF RND(1)<.4 THEN X1=INT(X1
    )+.21
270 PRINTX1;:GOSUB 12400:PRINT
    TAB(25);XY$:NEXT
280 :
```

```
9999 END
12400 Y9$="":X1$=RIGHT$(STR$(X1),
      LEN(STR$(X1))-1):X3=LEN(X1$
      ):X4=LEN(Y1$)
12405 FOR Y7=1 TO X4:IF MID$(Y1$,
      Y7,1)<>"." THEN NEXT:X5=Y7:
      GOTO 12415
12410 X5=Y7:Y7=X4:NEXT
12415 FOR Y7=1 TO X3:IF MID$(X1$,
      Y7,1)<>"." THEN NEXT:X6=Y7:
      GOTO 12430
12420 X6=Y7:Y7=X3:NEXT
12430 IF X6<=X5 THEN 12440
12435 XY$=MID$(X1$,X6-X5+1,X5-1):
      GOTO 12470
12440 IF X5<>X6 THEN FOR Y4=1 TO
      X5-X6:Y9$=Y9$+CHR$(32):NEXT
12450 XY$=Y9$+LEFT$(X1$,X6-1)
12470 IF X1=INT(X1) THEN 12490
12475 XY$=XY$+MID$(X1$,X6,LEN(Y1$
      )-X5+1)
12490 RETURN
12491 REM PRIMA DI GOSUB:
12492 REM Y1$:FORMATO
12493 REM X1:NUMERO DA TRATTARE
12494 REM XY$:CONTERRA' IL RISULT
      ATO
12499 REM NOME: PRINT USING
```

13300 Elaborazione di stringhe

(Per qualsiasi Commodore)

Questa routine implementa ben sei funzioni, ciascuna delle quali ha il compito di facilitare e rendere più completo il trattamento delle stringhe. Il Basic Commodore, infatti, è piuttosto scarno e le istruzioni dedicate alle stringhe sono poche e non ne consentono una gestione soddisfacente e completa.

Prima di effettuare il GoSub si deve assegnare alla variabile YX un valore compreso tra uno e sei che ha il compito di indicare la funzione richiesta.

Elenchiamo le funzioni a disposizione:

Inserimento: YX=1

Permette di inserire in una stringa principale (X1$) una sotto stringa (SS$), a partire da una determinata posizione (X1).

La variabile XX$ conterrà il risultato dell'elaborazione.

Esempio:
X1$="IO HO UN COMMODORE COMPUTER"
SS$="DRIVE E UN "
X1=9
Risultato:
XX$="IO HO UN DRIVE E UN COMMODORE COMPUTER"

Come avrete sicuramente intuito, la variabile X1 deve contenere la posizione del carattere della stringa master dopo il quale si vuole che venga inserita la substringa.

Sovrapposizione: YX=2

Questa funzione è analoga alla precedente, ma la sottostringa SS$, invece di essere inserita, si sovrappone alla stringa pricipale X1$ a partire dalla posizione indicata da X1. Il risultato dell'elaborazione sarà, come prima, contenuto in XX$.

Se nell'esempio della precedente funzione non si fosse operato un inserimento, ma una sovrapposizione, il risultato sarebbe stato:
XX$="IO HO UN DRIVE E UNCOMPUTER"
(senza lo spazio tra "UN" e "COMPUTER").

Place: YX=3

Permette di conoscere la posizione del primo carattere di una sottostringa (SS$) all'interno di una stringa master (X1$).

Al "ritorno" la variabile numerica X1 conterrà la posizione cercata.

Esempio:
X1$="ESEMPIO DELLA FUNZIONE PLACE"
SS$="UNZ"
Risultato:
X1=16

Se la substringa non è presente nella stringa master, al ritorno dalla subroutine X0$ conterrà "ERR".

Se invece tutto fila liscio, e questo vale anche per le altre funzioni, X0$ conterrà "OK".

Duplicate: YX=4

Questa funzione serve a duplicare la stringa X1$ per un numero X1 di volte. Il risultato sarà contenuto in XX$.

Esempio:
X1$="CHIP "
X1=4
Risultato:
XX$="CHIP CHIP CHIP CHIP"
Esempio:
X1$="HELLO"
X1=350

Risultato:
X0$="ERR"

L'errore segnalato in quest'ultimo caso è dovuto al fatto che la stringa avrebbe più di 255 caratteri e ciò, come è noto, non è lecito.

Si ottiene un errore (X0$="ERR") anche se X1 è minore di due, per i motivi che ben è facile immaginare.

Centratura: YX=5

Questa funzione visualizza X1$ (centrata in orizzontale) sulla riga in cui si trova il cursore al momento della chiamata.

X1$ non deve possedere più di 40 caratteri, pena la consueta segnalazione d'errore (X0$="ERR").

Print At: YX=6

Questa funzione gira solo sul Commodore 64, poichè si serve della routine "Plot" del Kernal, il Sistema Operativo. Se volete saperne di più, leggete gli articoli sul Kernal che C.C.C. ha pubblicato in passato.

I primi quattro caratteri di X1$ (che NON saranno stampati) devono contenere le coordinate della cella in cui si vuole che venga stampato il primo carattere della stringa.

Gli esempi chiariranno ogni dubbio.

Volendo stampare la parola "DRIVE" sulla nona riga a partire dalla colonna 16, si dovrà effettuare:
X1$="1609DRIVE"

Le colonne si indicano con un numero da 00 (zero zero) a 39, le righe con un valore da 00 (zero zero) a 24 (lo schermo è composto infatti da 25 linee, ciascuna delle quali possiede 40 caratteri).

N.B. il programma **non** effettua controlli sulle coordinate (per motivi di spazio): valori eccessivi provocheranno malfunzionamenti del sistema, dai quali si esce fortunatamente per mezzo di Run/Stop & Restore.

Il programma dimostrativo è molto semplice e non necessita di spiegazioni.

```
100 REM ESEMPIO D'USO
110 REM ELABORAZIONE DI STRINGH
    E
120 REM PER QUALSIASI COMMODORE
130 REM (LA FUNZIONE 6 FUNZIONA
     SOLO
140 REM SUL C-64)
150 :
160 INPUT "FUNZIONE YX (1/6)":Y
    X
170 INPUT "STRINGA X1$";X1$
180 INPUT "STRINGA SS$";SS$
190 INPUT "VALORE X1";X1
200 GOSUB 13300
210 PRINT"FUN.="YX:PRINT"X0$="X
    0$:PRINT"X1$="X1$
220 PRINT"X1="X1:PRINT"XX$="XX$
    :GOTO 160
300 :
9999 END
13300 X0$="OK":NX=LEN(X1$):ON YXG
      OTO 13310,13320,13330,13350
      ,13360,13370
13310 XX$=LEFT$(X1$,X1)+SS$+RIGHT
      $(X1$,NX-X1):RETURN:REM  IN
      SERIMENTO
13320 XX$=LEFT$(X1$,X1)+SS$+RIGHT
      $(X1$,NX-X1-LEN(SS$)):RETUR
      N:REM  SOVRAPP.
13330 FOR SJ=1 TO NX:IF MID$(X1$,
      SJ,LEN(SS$))<>SS$ THEN NEXT
      :X0$="ERR":RETURN
13340 X1=SJ:SJ=NX:NEXT:RETURN:REM
       PLACE
13350 IF NX*X1>255 OR X1<2 THEN X
      0$="ERR":RETURN
13355 XX$="":FOR X9=2 TO X1:XX$=X
      X$+X1$:NEXT:RETURN:REM  DUP
      LICA STRINGHE
13360 IF NX>40 THEN X0$="ERR":RET
      URN
13365 PRINT TAB(20-NX/2);X1$:RETU
      RN
13370 IF LEN(X1$)<5 THEN X0$="ERR
      ":RETURN
13372 POKE 783,PEEK(783) AND 254:
      POKE 781,VAL(MID$(X1$,3,2))
13380 POKE 782,VAL(LEFT$(X1$,2)):
      SYS65520:PRINTRIGHT$(X1$,NX
      -4):RETURN:REM  AT
13390 REM VAR. DA DEFIN. PRIMA DI
       GOSUB
13392 REM X1$,X1,SS$,YX (VEDI ART
      ICOLO)
13399 REM ELABORAZIONE DI STRINGH
      E
```

13400 Linee in bassa risoluzione

(Solo per C/64)

Con questa routine è possibile tracciare "linee" sullo schermo in bassa risoluzione, ovvero in modo testo.

Il risultato è senza pretese, ma può esser considerato come valido strumento per capire come si gestiscono le linee (anche in alta risoluzione).

Prima di effettuare il GoSub è necessario assegnare un valore alle seguenti variabili:

X1/Y1: contengono rispettivamente l'ascissa (range: 0-39) e l'ordinata (0-24) di uno degli estremi della linea.

X2/Y2: contengono rispettivamente l'ascissa e l'ordinata dell'altro estremo della linea.

X4: contiene il codice schermo dei caratteri (PokeCode) che formeranno la linea.

X6: contiene il colore dei caratteri che compongono la linea (secondo la solita tabella 0-15).

La linea, è superfluo dirlo, cancella tutto ciò che "trova" sul suo percorso e viene tracciata in un tempo più che accettabile.

La subroutine effettua un controllo sui valori delle coordinate: se non rispettano l'intervallo lecito, la linea **non** viene tracciata e la routine ritorna assegnando alla variabile X0$ il solito "ERR". In caso contrario conterrà "OK".

Volendo plottare solo un punto e non tracciare un'intera linea, è sufficiente dare ai due estremi le stesse coordinate (X1=X2, Y1=Y2).

Il Demo pubblicato offre quattro opzioni: un ventaglio, dieci linee casuali, un'ellisse e una sinusoide. Queste ultime sono un esempio di plottaggio di singoli punti.

```
100 REM ESEMPIO D'USO
110 REM LINEE IN BASSA RISOLUZI
    ONE
120 REM PER C/64
130 :
140 POKE 53281,7:POKE 53280,7:P
    OKE 646,6
150 PRINTCHR$(147)
160 PRINT TAB(11);"SCLEGLI"
170 PRINT:PRINT"1 ELLISSE1"
180 PRINT"2 VENTAGLIO"

190 PRINT"3 DIECI LINEE CASUALI
    "
200 PRINT"4 SINUSOIDE"
210 PRINT"5 RITORNO AL BASIC"
230 GET A$:IF A$<"1" OR A$>"5"
    THEN 230
240 PRINTCHR$(147):ON VAL(A$)GO
    TO 250,280,310,350,390
250 X4=81:X6=16*RND(1):FOR I=0
    TO 360 STEP 9
260 X1=SIN(I*π/180)*18+20:Y1=CO
    S(I*π/180)*10+12:X2=X1:Y2=Y
    1:GOSUB 13400:NEXT
265 POKE 198,0:WAIT 198,1:GOTO
    150
280 X4=42:X6=6:X2=19:Y2=24:FOR
    I=0 TO 39 STEP 4:X1=I:Y1=0:
    GOSUB 13400:NEXT
290 POKE 198,0:WAIT 198,1:GOTO
    150
310 X4=46:X6=2:FOR I=1 TO 10:X1
    =40*RND(1):X2=40*RND(1)
320 Y1=25*RND(1):Y2=25*RND(1):G
    OSUB 13400:NEXT:POKE 198,0:
    WAIT 198,1:GOTO 150
350 X4=45:X6=0:FOR I=0 TO 39:X1
    =I:Y1=SIN(I*20/100)*12+12
355 X2=X1:Y2=Y1:GOSUB 13400:NEX
    T
360 POKE 198,0:WAIT 198,1:GOTO
    150
390 SYS2048
400 :
9999 END
13400 X0$="OK":IF X1<0 OR X2<0 OR
     X1>39 OR X2>39 THEN X0$="E
    RR":RETURN
13410 IF Y1<0 OR Y2<0 OR Y1>24 OR
     Y2>24 THEN X0$="ERR":RETUR
    N
13420 YX=-(X1-X2)/40:YY=-(Y1-Y2)/
    40
13425 Y8=INT(1024+X1+INT(Y1)*40)
13430 Y9=INT(1024+X2+INT(Y2)*40)
13435 IF Y8>Y9 THEN 13470
13437 GOTO 13457
13440 POKE Y8,X4:POKE 54272+Y8,X6
13455 X1=X1+YX:Y1=Y1+YY
13457 Y8=INT(1024+X1+INT(Y1)*40)
13458 IF Y8>=Y9 THEN POKE Y8,X4:P
    OKE 54272+Y8,X6:RETURN
13465 GOTO 13440
13470 X7=X1:X1=X2:X2=X7:X7=Y1:Y1=
    Y2:Y2=X7:GOTO 10020
13491 REM X1,Y1..COORDINATE ESTRE
    MO 1
13492 REM X2,Y2..COORDINATE ESTRE
    MO 2
13493 REM X4.....CODICE SCHERMO C
    ARATT.
13494 REM X6.....COLORE DELLA LIN
    EA
13495 REM X1-X2 0...39 ** Y1-Y2 0
    ...24
13499 REM LINE... IN BASSA RISOLU
    ZIONE
```

Come realizzare l'enciclopedia e utilizzarla nei propri listati.

A i lettori che hanno acquistato per la prima volta questo numero di Commodore Computer Club, illustriamo qui di seguito, in breve, i vantaggi derivanti dalla raccolta proposta. Questa, a pensarci bene, è la versione "superiore" della rubrica "1 RIGA" e potrebbe anche denominarsi... "Una schermata"!

Oltre che utili per costituire un'enciclopedia, i brevissimi sottoprogrammi pubblicati su ogni numero, sono anche validissimi strumenti di studio per coloro che desiderano approfondire le proprie conoscenze del Basic, esaminando, senza fatica, particolari routine o insolite tecniche di programmazione.

- Dato che può esser "chiamata" più di una volta nel corso di un programma, nessuna routine contiene istruzioni del tipo DATA oppure DIM, allo scopo di non creare confusione col listato principale.
- Nessuna routine può far riferimento ad altre routine dell'enciclopedia.
- Nessuna routine può contenere variabili "banali" (A, A$, eccetera), ma solo variabili poco usate (X1$, X8, Y0%, eccetera).
- Ogni routine deve apparire, **per intero,** sullo schermo del computer e consentire, proprio per questo motivo, di essere esaminata comodamente.
- Ogni routine deve esser numerata secondo uno standard che ha la particolarità di esser ricordato facilmente:

Righe	Contenuto
XXY00	Prima riga del sottoprogramma
XXY89	Ultima riga utile del sottoprogramma
XXY90 REM	Prima riga di spiegazioni
XXY99 REM	Nome della subroutine

in cui XX sono due valori variabili da 10 a 63; Y è un carattere numerico compreso tra 0 e 9.

Qualsiasi subroutine, in altre parole, inizia con un numero, di cinque caratteri, che termina **sempre** con "00". La stessa subroutine, d'altra parte, ha l'ultima riga numerata con "99". Digitando, ad esempio: LIST 10800-10899
si avrà la certezza di veder apparire sullo schermo, **per intero,** la routine il cui nome si trova nella riga 10899.

Prima di accedere alla routine, è necessario assegnare, alle variabili indicate con REM da riga XXY89 a XXY98, particolari valori per il suo corretto funzionamento. Al "ritorno" una o più variabili conterranno il risultato dell'elaborazione.

In questo modo, per esser più chiari, è possibile simulare alcuni comandi di versioni Basic avanzate oppure, addirittura, creare nuove e inedite istruzioni. Ad esempio, il comando: SOUND 1,800,500
che, nel C-16, riproduce un suono di tonalità 800 tramite la voce 1 per la durata 500, potrebbe venir riprodotta, in un'ipotetica subroutine per il Commodore 64, con: X1=1:X2=800:X3=500:GOSUB12400
nell'ipotesi, ovviamente, che la routine in oggetto sia allocata da riga 12400 a 12499.

I listati pubblicati "girano" su ogni computer, salvo dove indicato diversamente.

E' ovvio che nel caso del Vic-20, (che, come è noto, ha uno schermo di soli 506 caratteri), le subroutine "universali" funzionano correttamente, ma non possono apparire per intero in una sola schermata.

Per quanto riguarda la digitazione, si tenga presente che sulla rivista, per motivi di chiarezza, i comandi e le istruzioni Basic sono separati tra loro da spazi bianchi. Nel digitare le linee di programma, pertanto, è opportuno ignorarli altrimenti si rischia di non restare in una sola schermata. Se, per esempio, leggete:

```
12100 X1=34:  X2  = SQR(X3) + LOG(X1)
```

digitate nel modo seguente:

```
12100 X1=34:X2=SQR(X3)+LOG(X1)
```

senza, cioè, alcun carattere di separazione tra comandi ed istruzioni.

Collaborazione dei lettori

La collaborazione dei lettori è gradita, purchè si provveda a inviare **almeno** tre sottoprogrammi per volta, su nastro, disco oppure output di stampante. I listati di routine che non rispettano lo standard adottato non potranno esser presi in considerazione.

Tutti i lavori pubblicati verranno compensati con prodotti della Systems Editoriale (cassette di programmi, libri, abbonamenti, copie arretrate, eccetera).

13500 Comandi fuori programma

(Solo per C/64)

Questa routine esegue le istruzioni Basic contenute nella variabile alfanumerica X1$, che **non** deve essere vuota nè contenere più di 39 caratteri (pena X0$="ERR").

Assegnando, per esempio, ad X1$ le istruzioni "POKE53281,1:CLR" ed effettuando il GoSub, lo schermo sarà colorato in bianco e le variabili verranno annullate.

Il sottoprogramma possiede anche un controllo di errore: se i comandi sono insensati, la routine interrompe il programma e rende noto l'errore che si è verificato.

La routine non contamina zone di schermo nè sposta il cursore. Funziona in un modo piuttosto insolito: viene creato uno pseudo-schermo a partire da 49152 in modo da non influenzare lo schermo principale. Lo pseudo-schermo non viene visualizzato ma, per il sistema operativo, è quello... ufficiale. Il resto è affidato alla ormai famosa tecnica del buffer di tastiera (C.C.C. ne ha parlato più di una volta...). Per non spostare il cursore si sono usate particolari routine del KERNAL, il sistema operativo del C/64.

Non è quindi possibile far eseguire comandi che implichino un qualsiasi Output sul video, come PRINT, INPUT, LIST ed altri.

Si possono tranquillamente utilizzare, al contrario, gli altri comandi come, ad esempio, LET, POKE e tutte le funzioni.

Il programma dimostrativo chiede una qualsiasi espressione numerica, proponendone una. Dopo aver premuto il tasto Return, il programma pone davanti all'espressione i due caratteri "A=" e chiama la subroutine. Al ritorno, come è evidente, "A" conterrà il risultato, che verrà subito visualizzato.

La seconda parte del Demo chiede un qualsiasi comando, come ad esempio "POKE53281,1" (digitate le virgolette prima e dopo perchè, all'interno, è presente una virgola, altrimenti viene segnalato Extra Ignored). Se non vi sono errori il comando sarà eseguito; in caso contrario viene comunicato quale errore si è verificato (provate a sbagliare di proposito...).

```
100 REM ESEMPIO D'USO
110 REM PER COMANDI FUORI PROGR
    AMMA
120 REM SENZA INFLUENZA SULLO S
    CHERMO
130 REM SOLO PER C/64
150 :
160 PRINTCHR$(147); TAB(10);"IN
    T(20+SQR(6000+400)+9↑2/[PI]
    )"
170 PRINTCHR$(19);:INPUT "ESPRE
    SS.";X1$
180 X1$="A="+X1$:GOSUB 13500
210 PRINT:PRINT"RISULTATO: ";A
220 PRINT:PRINT"PREMI UN TASTO"
230 POKE 198,0:WAIT 198,1:POKE
    198,0
240 X1$="":PRINTCHR$(147);:INPU
    T "COMANDO (ES.: POKE53281,
    1)";X1$
250 GOSUB 13500:PRINTX0$
260 IF X0$="ERR" THEN PRINT:PRI
    NT"TROPPE O NESSUNA LETTERA
    !":END
270 :
9999 END
13500 Y5=49232:X0$="OK":IF LEN(X1
      $)<1 OR LEN(X1$)>39 THEN X0
      $="ERR":RETURN
13505 POKE 783,PEEK(783) OR 1:SYS
      65520:POKE 830,PEEK(781):PO
      KE 831,PEEK(782)
13510 POKE 648,192:PRINTCHR$(147)
      ;X1$
13515 PRINT"[3 DOWN]GOTO13530":RE
      M ATTENZIONE A RINUMERARE
      13530
13520 POKE 631,19:POKE 632,13:POK
      E 633,13
13525 POKE 634,13:POKE 198,4:END
13530 POKE 648,4:IF PEEK(Y5)<>63
      THEN 13580
13535 PRINTCHR$(147);"QUESTE ISTR
      UZIONI NON SONO CORRETTE:"
13540 PRINT:PRINTX1$
13545 PRINT:PRINT"PROVOCANO INFAT
      TI IL SEGUENTE [RVS] ERRORE
       [RVOFF]:":PRINT
13550 FOR Y7=0 TO 39:POKE 1264+Y7
      ,PEEK(Y5+Y7):NEXT:POKE 198,
      0:X0$="ERR":END
13580 POKE 783,PEEK(783) AND 254:
      POKE 781,PEEK(830):POKE 782
      ,PEEK(831):SYS65520
13590 POKE 198,0:RETURN
13591 REM LA VARIABILE X1$ DEVE C
      ON-
13592 REM TENERE I COMANDI DA ESE
      GUIRE.
13599 REM NOME: COMANDI FUORI PRO
      GRAMMA
```

13600 OR esclusivo

(Solo per C/64)

Il Basic V2 del C/64 non offre la possibilità di effettuare questa operazione logica, disponibile soltanto in Assembler.

Questo sottoprogramma è infatti una brevissima routine in linguaggio macchina: permette di calcolare l'OR esclusivo tra due operandi i cui valori possono variare tra zero e 65535.

Si assegnano ad X1 e X2 i due operandi: si otterrà il risultato nella variabile numerica XY; valori di X1 oppure di X2 che non rispettano l'intervallo stabilito daranno luogo al consueto messaggio di errore (X0$="ERR").

E' d'obbligo però dare una descrizione di questo operatore logico binario; la "tabella della verità" è la seguente:

Input		Output
"A"	"B"	
0	0	0
1	1	0
1	0	1
0	1	1

Si noti che il risultato è unitario solo quando i due bit in Input sono opposti tra loro.

Questa operazione logica binaria è molto utile soprattutto per cambiare di stato uno o più bit.

Per commutare il terzo bit di un byte, ad esempio, è sufficiente compiere un EOR (Or Esclusivo) fra il byte in questione e una "maschera" che ha solo il bit 3 settato (eguale, cioè, ad 1).

Il programma dimostrativo commuta in continuazione il bit 7 di una cella video ottenendo il lampeggio del carattere che si trova in quella locazione.

Il bit 7 del codice Poke di ciascun carattere indica infatti se il carattere stesso deve essere visualizzato in positivo oppure in reverse (bit 7 settato = reverse, bit 7 nullo = positivo).

```
100 REM ESEMPIO D'USO
110 REM OR ESCLUSIVO (EOR)
120 REM PER C/64
130 :
140 PRINTCHR$(147) TAB(5)"LETTE
    RA LAMPEGGIANTE"
150 POKE 1024,8
160 FOR G=1 TO 200:NEXT
170 X1=PEEK(1024):X2=128:GOSUB
    13600
180 POKE 1024,XY:GOTO 160
190 :
9999 END
13600 Y1=679:REM START L.M. RILOC
      ABILE
13605 X0$="OK"
13610 IF X1<0 OR X1>65535 OR X2<0
       OR X2>65535 THEN X0$="ERR"
      :RETURN
13620 POKE Y1,169:POKE Y1+2,73:PO
      KE Y1+4,133:POKE Y1+5,251:P
      OKE Y1+6,169
13625 POKE Y1+8,73:POKE Y1+10,133
      :POKE Y1+11,252:POKE Y1+12,
      96
13630 Y2=INT(X1/256):Y3=X1-256*Y2
13635 Y4=INT(X2/256):Y5=X2-256*Y4
13640 POKE Y1+1,Y3:POKE Y1+3,Y5:P
      OKE Y1+7,Y2:POKE Y1+9,Y4:SY
      SY1:REM ATTENZIONE
13650 XY=PEEK(251)+PEEK(252)*256
13680 RETURN
13690 REM PRIMA DI EFFETTUARE IL
      GOSUB
13691 REM LE VARIABILI X1 ED X2 D
      EVONO
13692 REM CONTENERE GLI OPERANDI
13693 REM VERRA' EFFETTUATA L'OPE
      RAZIONE
13694 REM DI OR ESCLUSIVO (XY).
13695 REM (0,0)=0 (0,1)(1,0)=1 (1
      ,1)=0
13699 REM NOME: OR ESCLUSIVO
```

13700 Bit image su Mps 803/801

(Per qualsiasi Commodore)

Consente di disegnare in alta risoluzione sulle stampanti MPS 803 ed MPS 801 le quali, lo ribadiamo, sono completamente soft-compatibili.

Si assegna alla matrice X1$(X) l'immagine che si vuole mandare in stampa. X può variare tra zero e sei poichè la testina è formata da sette aghi.

I punti "accesi", ma è meglio dire che saranno stampati, devono essere rappresentati da un asterisco (*). I punti spenti, invece, possono essere rappresentati da un qualsiasi altro carattere.

Le sette varibili con indice devono avere tutte la medesima lunghezza, comunque **non** devono contenere più di 10 caratteri, pena il solito messaggio di errore (X0$="ERR").
Prima di effettuare il GoSub si deve però assegnare alla variabile numerica X2 uno dei seguenti valori:

X2=0: dopo la stampa in alta risoluzione, si ritorna in modo normale e viene effettuato un carriage Return.
X2=1: come prima, ma senza carriage Return.
X2>1: non si ritorna al modo normale, nè si effettua un ritorno di carrello, consentendo la stampa di più caratteri programmati allineati.

L'Output ed il listato stesso del programma dimostrativo dovrebbero spazzare via ogni dubbio: vengono stampati una greca (assegnando ad X2 un valore superiore ad uno, in questo caso tre) ed il Logo della Commodore (X2=1).

E' consigliabile, per comprendere meglio il funzionamento del sottoprogramma, consultare il manuale della stampante.

```
100 REM ESEMPIO D'USO
110 REM HIRES SU STAMPANTI COMM
    ODORE:
120 REM MPS803/801(E COMPATIBIL
    I)
130 REM PER QUALSIASI COMMODORE
140 :
150 PRINTCHR$(147):PRINT"ATTEND
    ERE 20"
155 :
160 X1$(0)="****.....*"
170 X1$(1)="....*...*."
180 X1$(2)="***..*.*.."
190 X1$(3)="***...*..."
200 X1$(4)="***..*.*.."
210 X1$(5)="....*...*."
220 X1$(6)="****.....*"
230 X2=3:FOR I=0 TO 20:GOSUB 13
    700:PRINTCHR$(19)I:NEXT
240 OPEN 4.4:FOR I=1 TO 5:PRINT
    #4,CHR$(15):NEXT:CLOSE 4
241 :
242 REM  ****** LOGO COMMODORE
    ******
243 :
250 X1$(0)="''**'''"
260 X1$(1)="'******"
270 X1$(2)="**''**'"
280 X1$(3)="**'''''"
290 X1$(4)="**''**'"
300 X1$(5)="'******"
310 X1$(6)="''**'''"
320 X2=1:GOSUB 13700:OPEN 4.4:P
    RINT#4," LOGO COMMODORE":CL
    OSE 4
900 :
9999 END
13700 X0$="OK":FOR X9=0 TO 6
13705 X1=LEN(X1$(0)):IF LEN(X1$(X
      9))<>X1 OR X1>10 THEN X0$="
      ERR":RETURN
13710 NEXT
13715 FOR X9=1 TO X1:X1(X9)=0:FOR
       X8=0 TO 6
13720 IF MID$(X1$(X8),X9,1)="*" T
      HEN X1(X9)=X1(X9)+2↑X8
13725 NEXT:X1(X9)=X1(X9) OR 128:N
      EXT
13735 OPEN 4,4:PRINT#4,CHR$(8);:F
      OR X9=1 TO X1:PRINT#4,CHR$(
      X1(X9));:NEXT
13736 IF X2=0 THEN PRINT#4,CHR$(1
      5):GOTO 13740
13737 IF X2=1 THEN PRINT#4,CHR$(1
      5);:GOTO 13740:REM (ALTRI V
      ALORI LASCIANO INVAR.)
13740 CLOSE 4:RETURN
13790 REM VARIABILI:
13791 REM X1$(X) = IMMAGINE DA ST
      AMPARE
13792 REM X DA 0 (ZERO) A 6 (7 RI
      GHE)
13794 REM X2=0 RIT. CARR. E MODO
      NORM.
13795 REM X2=1 MODO NORM. SENZA R
      ETURN
13796 REM X2 <> 0 ED 1.....INVARI
      ATO
13797 REM (VEDI ARTICOLO SU C.C.C
      .)
13799 REM HIRES SU MPS803/801
```

13800 Messaggi in extended Background Color Mode

(Per C/64)

Il circuito integrato video del C/64 (6567 detto anche: VIC II) è molto sofisticato e offre molteplici possibilità di visualizzazione. Quasi sempre però, a causa del limitato Basic residente, privo di istruzioni specifiche per questo circuito, i nostri programmi si limitano al modo testo.

Ecco alcune possibilità del circuito VIC II:
- modo testo semplice;
- modo testo multicolor;
- modo testo ext. background;
- caratteri definibili dall'utente;
- alta risoluzione bit map;
- bit map multicolor;
- gestione interrupt;
- bassa e alta risoluzione nello stesso istante (raster register);
- gestione di otto sprite dalle infinite possibilità (collisioni, movimenti ed altro);
- sprite multicolor;
- scrolling "fine";
- gestione di ben 16 sprite, pur se ricorrendo a qualche artificio.

Il tutto è relativamente complesso da programmare: sono infatti necessarie laboriosissime operazioni di Poke che aiutano, è vero, a conoscere meglio la macchina, ma certo non facilitano la vita al programmatore. Non parliamo poi della grafica bit-map, che necessita del linguaggio macchina per una gestione sufficientemente veloce.

La subroutine pubblicata serve a stampare messaggi sul video, i cui caratteri abbiano un colore di sfondo diverso dallo sfondo dello schermo (extended background color mode oppure, più semplicemente, EBCM). Tale tecnica si rivela molto utile nei programmi interattivi, quando è necessario richiamare l'attenzione dell'utente per riferire un messaggio di particolare importanza.

Il sottoprogramma **non** è comunque un mezzo sufficiente per utilizzare appieno l'EBCM, ma è senz'altro un valido strumento didattico.

Prima di effettuare il GoSub, come di consueto, alcune variabili devono contenere determinati valori che condizioneranno il funzionamento della routine, e precisamente:

X1$: è la stringa che sarà stampata sul video usando l'EBCM;

X5-Y5: contengono le coordinate (rispettivamente X e Y, la prima da 0 e 39, la seconda da 0 a 24) che determinano la posizione sul video in cui inizierà la stampa della stringa. Da notare che quando si chiama la subroutine lo schermo viene pulito con CHR$(147);

Y1-Y2: contengono rispettivamente il colore dello schermo (loc. 53281) e il colore del bordo (loc. 53280);

Y3: è il colore dei punti accesi, ovvero dei caratteri che compongono il messaggio (tabella 0-15);

Y4: contiene il colore dello sfondo dei caratteri;

XY: rappresenta il tempo (in secondi) durante il quale il messaggio resta sul video con l'EBCM attivato. Una volta scaduto il tempo, il cursore, lo schermo e il bordo riprenderanno i colori che avevano al momento della chiamata. Viene anche cancellato il video e restituito il controllo al programma principale.

N.B.: valori spropositati nelle variabili sopradescritte provocheranno, nei limiti del possibile, il messaggio di errore (X0$="ERR").

Il programma di esempio dell'uso visualizza la parola "ATTENDI" al centro dello schermo in EBCM. In seguito chiede tutti i parametri necessari per visualizzare un secondo messaggio e l'eventuale presenza di un errore viene comunicata. Ricordate che NON è possibile, in EBCM, visualizzare caratteri il cui Poke Code (codice schermo) sia maggiore di 63 (otterrete altri caratteri senza senso). Perchè?

Spiegazione di fantasia:

Non è giusto che si usufruisca di un'interessante particolarità e non se ne venga in qualche modo penalizzati.

Spiegazione logico - informatico - razionale:

L'argomento è relativamente complesso ma, come al solito, la spiegazione la troverete su qualche fascicolo di C.C.C.

```
  110 REM     MESSAGGI IN EXTENDED
  120 REM BACKGROUND COLOR MODE
  130 REM SOLO PER C/64
  140 :
  150 PRINTCHR$(147)
  160 PRINT" TRA POCO COMPARIRA'
      UN MESSAGGIO...."
  170 FOR I=0 TO 1000:NEXT
  180 X5=14:Y5=12:Y1=1:Y2=3:Y3=2:
      Y4=15:X1$=" ATTENDI ":XY=5:
      GOSUB 13800
  190 PRINT"ORA IMMETTI TU I VALO
      RI..."
  200 INPUT "X COORDINATA";X5
  210 INPUT "Y COORDINATA";Y5
  220 INPUT "COLORE DI SFONDO";Y1
  230 INPUT "COLORE DEL BORDO";Y2
  240 INPUT "COLORE DEI PUNTI ACC
      ESI";Y3
  250 INPUT "COLORE DEI PUNTI SPE
      NTI";Y4
  260 INPUT "MESSAGGIO";X1$
  270 INPUT "TEMPO DI PERMANENZA"
      ;XY
  280 GOSUB 13800
  290 PRINTCHR$(147);X0$
  300 :
 9999 END
13800 X7=255:Y8=53280:X0$="OK":IF
       X5<0 OR X5>39 OR LEN(X1$)>
      254 THEN X0$="ERR":RETURN
13805 IF Y5<0 OR Y5>24 THEN X0$="
      ERR":RETURN
13810 IF XY<0 THEN X0$="ERR":RETU
      RN
```

```
13815 IF Y1<0 OR Y2<0 OR Y3<0 OR
      Y4<0 OR Y1>X7 OR Y2>X7 OR Y
      3>X7 OR Y4>X7 THEN XY=-1:GO
      TO 13810
13825 PRINTCHR$(147):POKE 251,PEE
      K(Y8+1):POKE 252,PEEK(Y8):P
      OKE 253,PEEK(646)
13830 POKE 53265,PEEK(53265) OR 6
      4
13840 POKE 783,PEEK(783) AND 254:
      POKE 781,Y5:POKE 782,X5:SYS
      65520:X1$=CHR$(18)+X1$
13850 POKE Y8+1,Y1:POKE Y8,Y2:POK
      E 646,Y3:POKE Y8+3,Y4:PRINT
      X1$:Y9=XY*950
13855 FOR Y6=0 TO Y9:NEXT:POKE Y8
      +1,PEEK(251):POKE Y8,PEEK(2
      52):POKE 646,PEEK(253)
13860 POKE 53265,PEEK(53265) AND
      191:PRINTCHR$(147):RETURN
13890 REM X5-Y5 COORD/STAMPA X1$(
      =MESS.)
13891 REM Y1=SFONDO      Y2= BORDO
13892 REM Y3=SCRITTE     Y4= SF. SC
      RITTE
13894 REM XY=SECONDI CIRCA DI PER
      MANENZA
13899 NOME: MESSAGGI IN EXT. BACK
      GROUND
```

13900 Save & Load pagina video su/da disco

Una routine di uso estremamente semplice che, servendosi di files sequenziali, salva e carica schermate su floppy disk. Vengono salvati (oppure caricati):

- i caratteri (1000 byte).
- il colore dei caratteri (altri 1000 byte).
- il colore dello schermo (1 byte).
- il colore del bordo (1 byte).

Eventuali errori che si siano verificati durante le operazioni di caricamento/salvataggio (drive not ready, disk full, file exists ed altri) vengono immediatamente resi noti.

Prima di GoSub si deve porre in X1$ il nome del file/schermo che si desidera salvare (caricare) che **non** deve essere più lungo di sedici caratteri: nel caso questa variabile stringa sia vuota, il file viene automaticamente chiamato "Schermata".

Alla chiamata della routine, X1 deve contenere "1" (uno) se si desidera caricare una schermata, un qualsiasi altro valore ne provocherà il salvataggio.

E' bene ricordare che ciascun file/schermo occupa sempre otto blocchi.

```
110 REM VIDEO SU DISCO
140 POKE 53281,1:POKE 53280,1:P
    OKE 646,14:PRINTCHR$(147)
150 FOR I=1 TO 8:PRINT"DEMO ";:
    NEXT
160 PRINT:PRINT:PRINTCHR$(144)"
    ORA QUESTA SCHERMATA DIMOST
    RATIVA"
170 PRINT:PRINT" VIENE SALVATA
    SU DISCO (NOME: HELLO)"
180 PRINTCHR$(31)"ATTENDI"
190 FOR I=1704 TO 1903:POKE I,2
    56*RND(1):POKE 54272+I,16*R
    ND(1):NEXT
200 X1=0:X1$="HELLO":GOSUB 1390
    0:POKE 53280,0:POKE 53281,0
210 SYS65409:PRINTCHR$(147)"ORA
     VIENE CARICATA...":X1=1:X1
    $="HELLO":GOSUB 13900
9999 END
13900 IF LEN(X1$)=0 THEN X1$="SCH
      ERMATA"
13905 OPEN 8,8,15,"I0":ON X1GOTO
      13950
13910 OPEN 9,8,2,X1$+",S,W":GOSUB
       13985
13925 PRINT#9,CHR$(PEEK(53281));:
      PRINT#9,CHR$(PEEK(53280));
13930 FOR Y5=1024 TO 2023:PRINT#9
      ,CHR$(PEEK(Y5));:GOSUB 1398
      5
13935 PRINT#9,CHR$(PEEK(Y5+54272)
      );:NEXT:CLOSE 9:CLOSE 8:RET
      URN
13950 OPEN 9,8,2,X1$+",S,R":GOSUB
       13985:GET #9,X2$,X3$:IF LE
      N(X2$)=0 THEN X2$="0"
13956 POKE 53281,ASC(X2$):IF LEN(
      X3$)=0 THEN X3$="0"
13960 POKE 53280,ASC(X3$):FOR Y5=
      1024 TO 2023:GET #9,X2$,X3$
      :IF LEN(X2$)=0 THEN X2$="0"
13962 IF LEN(X3$)=0 THEN X3$="0"
13965 POKE Y5,ASC(X2$):POKE Y5+54
      272,ASC(X3$):NEXT:CLOSE 9:C
      LOSE 8:RETURN
13985 INPUT#8,X3,X4$:IF X4$="OK"
      THEN RETURN
13989 SYS65409:PRINTX3;X4$:END
13991 REM X1$=NOME. X1=1:LOAD; X1
      =0:SAVE
13999 REM NOME:VIDEO DA/SU DISCO
```

Stampare o non stampare? Visualizzare, forse...

A che cosa può servire la stampante? Quali sono le principali caratteristiche della Mps 803 e delle sue compatibili?

di Maurizio Dell'Abate

Per avere un'idea chiara sulla Mps 803 è bene conoscere, almeno a grandi linee, la storia delle stampanti Commodore degli ultimi anni.

La prima (o quasi) stampante a livello hobbistico presentata sul mercato fu la 1525, ai tempi del Vic 20. Poco dopo nacque il Commodore 64, destinato ad essere, negli anni successivi, il leader dei personal della sua fascia.

Tutto il software per questo computer (Easy Script, Print Shop, Word Pro III, Simons' Basic, Koala Printer, 120 colonne, eccetera) fu sviluppato in modo da funzionare correttamente con la 1525 e con la successiva stampante Commodore, siglata Mps 801. Questa presentava, rispetto alla precedente, lievi miglioramenti tecnici, poichè era stata progettata tenendo conto dell'esigenza di conservare la compatibilità con tutto il software già sviluppato.

Le prestazioni della Mps 801 non erano, però, eccellenti, tanto che si sentì il bisogno di progettare una stampante migliore, più robusta, con buona qualità di scrittura e velocità di stampa adeguata.

Nacque così la Mps 802, destinata, purtroppo, a scomparire precocemente dal mercato a causa della completa incompatibilità con buona parte del software disponibile al momento del "lancio": qualcuno, in casa Commodore, dimenticò, forse, che l'utente non compra una periferica che non può usare...

Ed ecco infine la recente (o quasi) comparsa della Mps 803, stampante che offre migliori prestazioni e un funzionamento ineccepibile con la massima parte del software esistente, grazie al modo di funzionare che, dal punto di vista software, è identico alla Mps 801.

La stampante per il neofita

Per aiutare quanti non hanno mai sentito parlare di stampanti, è d'obbligo chiarire i principi di funzionamento di questa periferica ed elencarne i possibili usi.

La stampante è un mezzo per trasferire su carta (e produrre, cioè, documenti durevoli e trasportabili) i dati elaborati dal computer cui viene connessa: possono essere listati di programmi, elaborati di word processor, archivi, output di programmi gestionali, grafici e disegni di vario tipo.

L'immagine è prodotta da una testina formata da un certo numero di aghi incolonnati (sette, nel caso della Mps 803) che, trascinata da un motore, stampa i caratteri sulla carta. Si nota facilmente, infatti, che ogni carattere o disegno è formato da più righe verticali, formate, ciascuna, da piccoli puntini posti l'uno sull'altro (dot).

La qualità di stampa, com'è intuibile, è tanto più alta quanto maggiore è il numero degli aghi presenti nella testina. Nel nostro caso... un attimo di pazienza!!

Le principali caratteristiche

La Mps 803 è caratterizzata da dimensioni relativamente ridotte (cm 33x19x7: peso: 2 kg circa) e da un aspetto gradevole. E' disponibile in tre colori: nero (per C/16 e Plus/4), caffelatte (per C/64 e Vic 20) e, recentemente, anche in bianco, per affiancarla al nuovo Commodore 128. La differenza, beninteso, si limita soltanto al colore e nulla vieta, a un possessore di C/64, di comprarla nera, se la desidera!

Il trascinamento della carta è del tipo a frizione, come quello di una normale macchina per scrivere: accetta di conseguenza sia il foglio singolo che il modulo continuo. Per ottenere, tuttavia, una maggior precisione con il modulo continuo, è disponibile sul mercato un apposito accessorio (opzionale), il cosiddetto "trattore" dal prezzo contenuto, intorno alle 30 mila lire.

La velocità di stampa è accettabile (60 CPS, Characters Per Second) ed è del tipo "bidirezionale ottimizzato": ciò significa che la testina stampa sia quando va da sinistra a destra, che quando ritorna (a differenza della Mps 801). Il listato di un programma di lunghezza pari a 5 kbytes viene stampato in circa tre minuti, tempo più che accettabile.

Il carattere è composto da una matrice di 7 (altezza della testina) per 6 dot, come quella della Mps 801. La testina ha però dimensioni diverse che contribuiscono a conferire al carattere una forma più gradevole e meglio leggibile. La qualità di stampa diminuisce notevolmente in modo minuscolo, a causa della mancanza di "discendenti", caratteristici delle lettere come la "p", la "g", la "y", ed altre. In questi casi, infatti, il carattere viene "sollevato", ma il risultato è comunque accettabile.

E' bene notare che la 803 è in grado di stampare tutti i caratteri semi-grafici Commodore, vale a dire i "geroglifici" serigrafati sul frontale dei tasti.

Altre caratteristiche tecniche

E' possibile, con la Mps 803, disegnare in alta risoluzione (bit image mode), unendo al testo grafici o tabelle di genere diverso. La stampante può così, emulando il plotter, trasformarsi in uno strumento dalle infinite possibilità, pur avendo applicazioni hobbystiche: tutto sta nel possedere il software adeguato.

Con la Mps 803 si hanno a disposizione solamente tre modi di stampa: normale, reverse ed "elongato" (doppia larghezza); non è possibile il modo Near Letter Quality (NLQ), il corsivo o il grassetto ma, in compenso, utilizzare la 803 è estremamente semplice e può portare a risultati più che soddisfacenti in ogni caso.

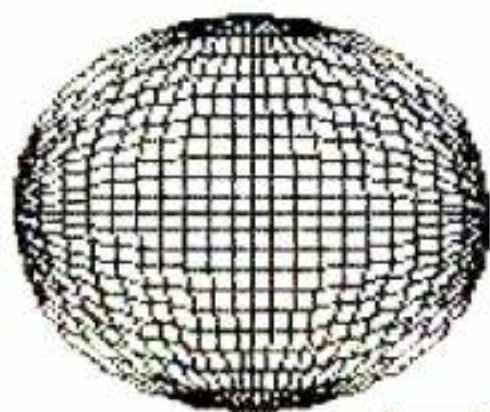
Il numero massimo di caratteri per rigo è fissato in 80, valore ideale per la maggior parte delle applicazioni.

Esistono tuttavia appositi programmi capaci di stampare fino 120 caratteri per rigo (e scusate se è poco...).

L'interlinea può essere impostata via software oppure via hardware, tramite il primo dei due deviatori posti sul retro della macchina. Il secondo deviatore presente serve ad impostare il numero di device che può assumere valore quattro o cinque.

Oltre all'interruttore di alimentazione, posto sulla fiancata destra, è presente un unico pulsante sul frontale: è il solito "paper advance", che serve per fare avanzare la carta senza girare manualmente il rullo. La pressione di questo tasto durante l'accensione provoca il self—test della macchina.

La spia led rossa, quando è accesa, segnala lo stato di ON della stampante; quando lampeggia, invece, vuole segnalare che si è verificato un errore, come l'assenza di carta; un cicalino non avrebbe tuttavia guastato...

La Commodore assicura una durata della testina pari ad un minimo di venti milioni di caratteri: sarà vero?

Le "compatibili"

In seguito al successo della Commodore, e del C/64 in particolare, molte fabbriche di stampanti si sono buttate a capofitto nel produrre modelli collegabili ai computer Commodore. In effetti qualsiasi stampante si può collegare grazie a particolari interfacce (RS-232, IEE-488). Purtroppo, però, sono totalmente incompatibili col software presente sul mercato.

Nascono, così, le stampanti 803-compatibili. Con questo termine si deve intendere la completa compatibilità con la stampante "ufficiale" della Commodore. Una stampante commercializzata con la sigla "803-compatibile" deve essere in grado di sostituire completamente l'originale.

Su questa stessa rivista sono state presentate, in precedenza, diversi modelli realmente compatibili con la stampante in questione.

Il nastro inchiostrato

La stampante ha bisogno, ovviamente, di inchiostro; questo impregna un nastro contenuto all'interno di una cartuccia, detta cartridge, di dimensioni ridotte che si inserisce agevolmente nell'apposito alloggiamento sollevando il coperchio della stampante. Il nastro è disponibile soltanto nel colore nero.

La sua durata, purtroppo, non è eccessiva (l'eufemismo è d'obbligo): appena acquistato, infatti, la stampa è ben marcata e di buona qualità. Successivamente si assesta sul grigio, rendendo necessario il ricambio della cartuccia. Il prezzo della cartuccia è relativamente ridotto (si può trovare anché a 10.000 lire). Qualche problema presenta la reperibilità, nonostante la Mps 803 sia tuttora la sola stampante Commodore a livello hobbystico presente sul mercato.

Documentazione

Il manuale fornito insieme con la macchina è sintetico ma esauriente e completo. Consta di 56 pagine solo in lingua inglese. Sarebbe bello se tutte le macchine Commodore fossero sempre accompagnate da documentazione completa, chiara, esauriente e, soprattutto, in italiano.

Tiriamo le somme

Spariamo subito il prezzo: lire 490.000 IVA esclusa, è la cifra presente sul listino ufficiale Commodore. Tuttavia abbiamo avuto occasione di vedere in qualche vetrina la Mps 803 (trattore e I.V.A. compresi) intorno alle 400.000 lire.

Un prezzo senza dubbio contenuto in rapporto alle prestazioni della macchina che, unita ad un drive e ad un buon word processor, consente (perchè no?) applicazioni semi—professionali. La stampante, lo ricordiamo, è anche un mezzo prezioso per stampare grafici o listati di programma.

La Mps 803 e le sue compatibili si collegano direttamente al C/64, C/128, C/16, Plus/4, Vic 20 attraverso il connettore seriale. Non è necessaria alcuna interfaccia dato che è sufficiente il cavetto fornito nella confezione.

Per ulteriori informazioni:

Commodore Italiana
Via F.lli Gracchi, 48
20092 Cinisello Balsamo (MI)
Tel: 02/61.83.22.45

I CARATTERI DELLA MPS 803

PICA MAIUSCOLO - ABCDEFGHIJKLMNOPQRSTUVWXYZ

Pica minuscolo - abcdefghijklmnopqrstuvwxyz

ELONGATO MAIUSCOLO - ABCDEFG...

elongato minuscolo - abcdefg...

REVERSE - [illegible]

PUNTEGGIATURA E VARI - !"#$%&'()+-£@*↑[]:;?,.<> 1234567890

PIU' TUTTI I CARATTERI GRAFICI

Come disegnare con il plotter 1520

Un brevissimo listato, suscettibile di notevoli sofisticazioni, per utilizzare senza problemi la piccola periferica della Commodore.

di Michele Maggi

Il plotter 1520 è una periferica piuttosto diffusa ma, per uso amatoriale od hobbistico, non trova una ben precisa applicazione a causa delle sue limitazioni.

Il terreno più fertile per un suo utilizzo rimane quello dei diagrammi e dei grafici e non certo quello della stampa di listati oppure di testi a causa della notevole lentezza.

Grazie al suo basso costo, tuttavia, molti utenti lo hanno acquistato ma, per i motivi anzidetti o non lo usano affatto oppure non ne sfruttano a fondo le caratteristiche.

Il programma che segue è rivolto proprio a coloro i quali, delusi delle modeste capacità della periferica, non sanno che nella realtà può rivelarsi molto utile in più occasioni.

Il programma

Si tratta di un programma che consente di disegnare con il Plotter agendo sui tasti cursore del computer.

Vengono offerte le opzioni di selezione del colore e di spostamento da un punto all'altro del foglio con il pennino sollevato oppure abbassato.

E' un programma sicuramente utile per coloro che hanno necessità di disegnare grafici senza tanti problemi e non intendono litigare con il manuale di istruzioni.

Il programma si basa sull'utilizzo degli indirizzi secondari del Plotter. Questi, come noto, rappresentano i diversi comandi che è possibile inviare al 1520.

Gli indirizzi secondari sono i seguenti: Zero (0). E' il valore di default, ovvero quello che automaticamente viene assegnato non appena si accende l'apparecchio. Con tale impostazione il plotter trascrive su carta i caratteri alfanumerici così come provengono dal computer.

Per esempio, il comando OPEN 4,6,0 "apre" il canale di comunicazione per il plotter. Aggiungendo l'istruzione PRINT# 4, "Commodore Computer Club" il messaggio verrà trascritto su carta. Il numero zero, valore di default, può venire omesso.

L'indirizzo "1" permette al Plotter di disegnare con linee e punti e di controllare il movimento della penna.

L'indirizzo "2" seleziona la penna imprimendo uno scatto al revolver portapenne.

L'indirizzo "3" seleziona le dimensioni del carattere.

Gli indirizzi 4, 5 e 6 non vengono analizzati, dato che non sono usati nel programma pubblicato.

L'indirizzo 7 resetta il Plotter simulando lo spegnimento e la riaccensione della periferica. Per esempio, le due istruzioni OPEN 7,6,7:PRINT# 7 faranno disegnare al Plotter i quattro quadratini che vengono disegnati al momento dell'accensione, riporteranno tutti i valori (relativi al colore della penna, alla grandezza del carattere, eccetera) ai valori standard.

Come funziona il programma

Una volta dato il RUN il Plotter viene resettato e, subito dopo, vengono richiesti il modo e il colore in cui si desidera operare.

Ciò significa che dovrete premere il tasto "M" (muovi) oppure "D" (disegna).

Nel primo caso (M) sarà possibile spostare il pennino (tenendolo sollevato) in corrispondenza di un punto qualsiasi della carta.

Nel secondo (D), invece, il pennino, messo in contatto con la carta, si sposterà in una delle quattro direzioni.

In entrambi i casi il movimento verrà impresso attraverso la pressione dei tasti del cursore in modo identico a quello in cui viene spostato il cursore dello schermo.

Effettuata in modo analogo la scelta del colore (premendo uno dei tasti "N", "B", "V" oppure "R"), si deve premere il tasto "E" (Exit) che permetterà, finalmente, di spostare la penna utilizzando i tasti cursore.

In qualsiasi momento sarà possibile cambiare le opzioni premendo il tasto dell'asterisco (*).

Se durante l'esecuzione del disegno si preme il tasto Return, la penna tornerà all'estremità sinistra dell'asse orizzontale.

Su noti che la velocità di spostamento è limitata ad un puntino per volta. Se, però, premete il tasto più (+), i segmenti tracciati avranno una lunghezza dieci

volte maggiore. Per tornare al tratto minore, è sufficiente premere il tasto meno (—).

Struttura del programma

La linea 170 apre il canale relativo al plottaggio in termini di coordinate X, Y e il canale relativo ai colori delle penne.

La linea 200 resetta il Plotter.

La linea 210 manda alla subroutine 350 che fissa il modo (disegno oppure plottaggio) e i colori.

Le linee 220-300 si occupano di stabilire i movimenti da effettuare oppure le linee da tracciare e, se viene premuto il tasto di asterisco (*) consentono il ritorno al menu.

Le linee 330 e 340 disegnano e spostano la penna.

Analisi dei comandi possibili

In questo programma vengono utilizzati solo due dei sei comandi di plottaggio che abbiamo a disposizione.

Questi sono: "M", che muove il pennino alle coordinate specificate, e "D" che disegna.

Gli altri comandi, non usati nel listato di queste pagine, sono:

"H" che muove verso il punto di origine assoluto.
"I" che stabilisce un punto di origine relativo alla posizione attuale.
"R" che muove verso le coordinate "X" e "Y" che si riferiscono, però, al punto di origine relativo fissato da "I".
"J" che sposta la penna in modo identico a "R", ma col pennino abbassato sulla carta.

Pensate di esser bravi?

Consigliamo ai nostri lettori di realizzare, sulla falsariga di quello che compare in queste pagine, un programma più versatile.

E' infatti possibile aggiungere comandi in grado di tracciare circonferenze, rettangoli, ellissi nei puni più disparati del foglio.

Non si dimentichi, inoltre, la possibilità di trascrivere messaggi oppure, per chi è veramente bravo, disegni memorizzati precedentemente su disco o nastro.

Se ritenete di aver realizzato un programma interessante, telefonate in Redazione (02/8467348) in modo da concordare la pubblicazione del vostro lavoro.

```
100 REM COME DISEGNARE COL POTT
    ER 1520
110 REM E QUALSIASI COMPUTER CO
    MMODORE
120 :
130 REM BY MICHELE MAGGI
140 :
150 W$-CHR$(18):Q$=CHR$(147):T$
    -CHR$(19)
160 X$-CHR$(147)+W$+"SCEGLERE L
    'OPZIONE"+CHR$(146)+"
      "
170 OPEN 1,6,1:OPEN 2,6,2
180 M$-W$+" MUOVI ":D$-W$+"DISE
    GNA":N$-W$+"NERO "
190 B$-W$+" BLU ":V$-W$+"VERDE"
    :R$-W$+"ROSSO":W-1
200 OPEN 7,6,7:PRINT#7:CLOSE 7
210 PRINTX$:GOSUB 350
220 GET A$:IF A$-"" THEN 220
230 IF A$-"+" THEN W-10:PRINTCH
    R$(19) TAB(26)A$:GOTO 220
240 IF A$-"-" THEN W-1:PRINTCHR
    $(19) TAB(26)A$:GOTO 220
250 IF A$-CHR$(145) THEN Y-Y+W
260 IF A$-CHR$(17) THEN Y-Y-W
270 IF A$-CHR$(29) THEN X-X+W
280 IF A$-CHR$(157) THEN X-X-W
290 IF A$-"*" THEN 210
300 IF A$-CHR$(13) THEN X-0:
310 PRINT#2,COL:ON ZGOSUB 330,3
    40
320 GOTO 220
330 PRINT#1,"M",X,Y:RETURN
340 PRINT#1,"D",X,Y:RETURN
350 PRINTCHR$(19):PRINT"MODO: "
    :PRINT"COLORE: ":PRINT
360 PRINT"M UOVI":PRINT"D ISEGN
    A":PRINT
370 PRINT"N ERO":PRINT"B LU":PR
    INT"V ERDE"
380 PRINT"R OSSO":PRINT:PRINT"E
     XIT"
390 GET A$:IF A$-"" THEN 390
400 IF A$-"M" THEN Z-1:PRINTT$:
    PRINT TAB(9)M$::GOTO 390
410 IF A$-"D" THEN Z-2:PRINTT$:
    PRINT TAB(9)D$:GOTO 390
420 IF A$-"N" THEN COL-0:PRINTT
    $:PRINT:PRINT TAB(9)N$:GOTO
     390
430 IF A$-"B" THEN COL-1:PRINTT
    $:PRINT:PRINT TAB(9)B$:GOTO
     390
440 IF A$-"V" THEN COL-2:PRINTT
    $:PRINT:PRINT TAB(9)V$:GOTO
     390
450 IF A$-"R" THEN COL-3:PRINTT
    $:PRINT:PRINT TAB(9)R$:GOTO
     390
460 IF A$-"E" THEN :PRINTT$W$"P
    REMERE * PER LE OPZIONI":RE
    TURN
470 GOTO 390
```

computer service

VENDITA PER CORRISPONDENZA

ACCESSORI PER COMPUTER COMMODORE

GRUPPO CONTINUITÀ
Fornito senza le 12 batterie a stilo ricaricabili. Consente il funzionamento del Vostro computer Commodore C64 o VIC 20 in assenza di corrente. Durata di funzionamento 30 minuti. Ricarica tramite alimentatore Commodore.

KIT ALLINEAMENTO TESTINA
Composto dal cacciavite, nastro di controllo e strumento di taratura con monitor audio permette il perfetto allineamento dei registratori digitali anche con nastri commerciali.

VELOCIZZATORE DI CARICAMENTO FLOPPY
Cartridge con un insieme di utility residenti su ros per velocizzare il drive nel Commodore 64.

INTERFACCIA RADIO
Indispensabile per registrare con registratore Commodore modello "C2N" i programmi speciali per computer trasmessi dalle emittenti radio.

CUFFIA PER COMMODORE C 64
Leggerissima permette l'ascolto personale del computer evitando di distrurbare durante i giochi.

COPIATORE PROGRAMMI
Dispositivo hardware per effettuare copie di nastri protetti o turbo utilizzando due registratori Commodore o compatibili.

DUPLICATORE CASSETTE
Indispensabile per realizzare delle copie, con un registratore normale, di un nastro protetto o con caricamento turbo

Articolo	Codice	Prezzo
Bus quadrislot	Art. CD 100	L. 55.000
Interfaccia cassette	Art. CD 101	L. 30.000
Duplicatore cassette	**Art. CD 102**	L. 30.000
Copiatore programmi	**Art. CD 103**	L. 30.000
Interfaccia radio	**Art. CD 104**	L. 30.000
Kit allineamento testina	**Art. CD 105**	L. 47.000
Alimentatore per C64 e VIC 20	Art. CD 106	L. 45.000
Gruppo continuità (fornito senza le 12 batterie a stilo ricaricabili)	**Art. CD 107**	L. 66.000
Pacco batterie (12 stilo 1,2 Volt ricaricabili	Art. CD 117	L. 52.000
Commutatore antenna TV/computer	Art. CD 108	L. 9.500
Tasto reset	Art. CD 109	L. 5.500
Interfaccia Centronics	Art. CD 112	L. 104.000
Espansione di memoria per C 16	Art. CD 114	L. 158.000
Velocizzatore di caricamento flop,	**Art. CD 115**	L. 49.000
Espansione di memoria per VIC 20 16K	Art. CD 116	L. 112.000
Modulatore Executive	Art. CD 120	L. 72.000
Penna ottica grafica	Art. CD 121	L. 45.000
Tavoletta grafica	Art. CD 130	L. 238.000
Multipresa con filtro - 2 prese	Art. CD 140	L. 41.000
Cuffia per Commodore C 64	**Art. CD 150**	L. 19.000
Stabilizzatore elettronico di tensione 500 W	Art. CD 160	L. 430.000
Gruppo di cintinuità 60 W	Art. CD 170	L. 400.000
Gruppo di continuità 200 V	Art. CD 180	L. 802.000
Inventer 12 Volt cc. 220 Volt ca. 100 Watt	Art. CD 190	L. 297.000
Cavo alimentazione	Art. CD 200	L. 4.600
Cavo drive o stampante Commodore	Art. CD 205	L. 8.500
Prolunga per Joystick - mt. 3	Art. CD 210	L. 25.000
Prolunga per cavo TV - mt. 3	Art. CD 215	L. 12.500
Cavo audio - mt. 6	Art. CD 220	L. 15.500
Adattatore Joistick (Atari e C64 al C 16)	Art. CD 225	L. 10.500
Adattatore registratore per C 16	Art. CD 226	L. 19.500
Mascherina antiriflesso 12"	Art. CD 300	L. 35.000
Nastro inchiostrato per Tally - mt. 80	Art. CD 610	L. 16.500
Nastro inchiostrato per Tally - mt. 180	Art. CD 611	L. 16.500
Nastro inchiostrato per Tally 1000 e Honeywell	Art. CD 612	L. 9.500
Nastro inchiostrato per Commodore MRS 801	Art. CD 614	L. 13.000
Nastro inchiostrato per Commodore MPS 802	Art. CD 616	L. 18.000
Nastro inchiostrato per Commodore MPS 803	Art. CD 618	L. 19.500
Mause per Commodore C 64	Art. CD 860	L. 240.000
Pacco carta lettura facilitata 24" x 11" modulo da 500 fogli con bordi a strappo	Art. CD 630	L. 13.500
Supporto stampante porta carta in plexiglass "fume" - normale	Art. CD 660	L. 59.000
Supporto stampante porta carta in plexiglass "fume" - rinforzato	Art. CD 670	L. 80.000
Floppy disk 5" singola faccia doppia densità "ODP" - conf. 10 pezzi	Art. CD 700	L. 40.000
Floppy disk 5" singola faccia doppia densità "CBS" - conf. 10 pezzi	Art. CD 702	L. 38.000
Floppy disk 5" singola faccia doppia densità "VERBATIM" - conf. 10 pezzi	Art. CD 704	L. 42.000
Floppy disk 5" singola faccia doppia densità "DYSAN" - conf. 10 pezzi	Art. CD 706	L. 68.000
Nastri magnetici C 10 digitali - conf. 10 pezzi	Art. CD 712	L. 20.000
Nastri magnetici C 15 digitali	Art. CD 714	L. 21.000
Copritastiera in plexiglass per C64 - C16 e VIC 20	Art. CD 750	L. 16.000
Copritastiera in stoffa per C64 - C16 e VIC 20	Art. CD 760	L. 10.500
Vaschetta portafloppy in plexiglass per 40 dischi con chiave	Art. CD 770	L. 30.000
Vaschetta portafloppy in plexiglass per 90 dischi con chiave	Art. CD 780	L. 37.000
Kit pulizia testine registratore	Art. CD 815	L. 13.500
Kit pulizia disk drive	Art. CD 820	L. 26.000
Kit pulizia tastiera	Art. CD 830	L. 16.500
Foratore disk in plastica (per utilizzare la seconda faccia dei dischi)	Art. CD 840	L. 10.000
Foratore disk in metallo "tako"	Art. CD 849	L. 14.000
Joystick Spectravideo II	Art. CD 850	L. 27.000
Joystick a Microswitch	Art. CD 851	L. 52.500
Joystick senza fili con unità ricevente (funziona a batteria)	Art. CD 852	L. 98.000
Joystick per Commodore 16 (originale)	Art. CD 130	L. 29.500

TUTTI I PREZZI SONO COMPRENSIVI DI IVA
NON SI ACCETTANO ORDINI INFERIORI A L. 30.000
CONTRIBUTO FISSO SPESE DI SPEDIZIONE L. 5000

SI ACCETTANO ANCHE ORDINI TELEFONICI AI NUMERI 0522/661647-661471

BUONO DI ORDINAZIONE

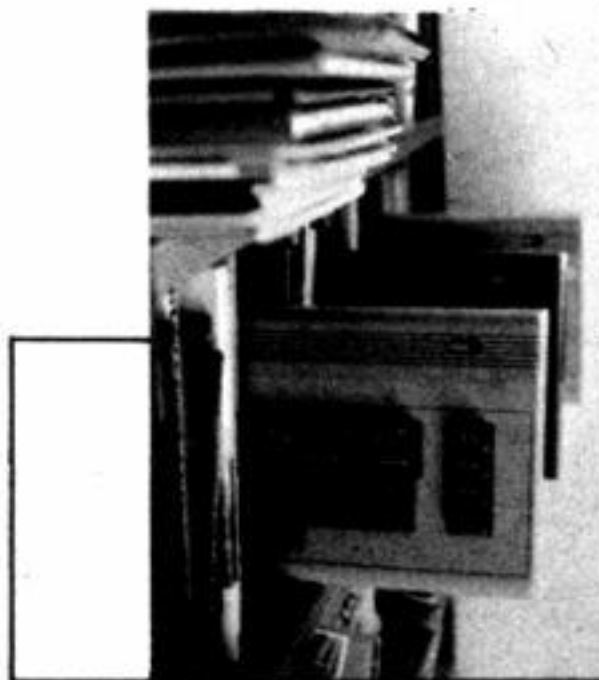
NOME - COGNOME

INDIRIZZO — N.

C.A.P. — CITTÀ — PROVINCIA

VOGLIATE INVIARMI IN CONTRASSEGNO

N.	Art.	L.
N.	Art.	L.
N.	Art.	L.
SPESE SPEDIZIONE		L. 5.000
PAGHERÒ AL POSTINO		L.

COMPUTER SERVICE VIA A. MANZONI, 49 - 42017 NOVELLARA (RE) - TEL. (0522) 661647

recensioni LIBRI

Guida di riferimento al C/128

Con l'uscita del C/128 sul mercato italiano era logico aspettarsi l'introduzione di libri sull'argomento. E' il caso di questa "Guida al C/128" della EVM.

Come ormai tutti sapranno, il C/128 dispone di tre modi di funzionamento (C/128, C/64 e CP/M) ma, di questi, il testo in oggetto tratta solo il primo.

Tale scelta ci è sembrata valida, sia perchè permette di inserire più informazioni sullo stesso argomento, sia perchè non costringe all'acquisto di un libro che, pur se completo risulterebbe voluminoso e costoso in conseguenza...

Alla fine del volume, comunque, viene proposto l'acquisto della "Guida al C/64", della stessa EVM, a condizioni vantaggiose, e nell'introduzione viene assicurata la prossima commercializzazione di un libro sul CP/M.

Esaminiamo ora i contenuti del volume che, diviso in sei capitoli, guida il lettore dalla programmazione più semplice alla complessa gestione del linguaggio macchina (LM) e del sistema operativo (SO).

Il primo capitolo, oltre a cenni di informatica generale e ad una breve spiegazione del formato delle linee Basic, presenta un elenco completo delle istruzioni del Basic 7.0 del C/128. Questa sezione è molto ben curata: ogni informazione sui singoli comandi comprende la loro corretta sintassi, una dettagliata descrizione ed esempi di utilizzo. Per avere un'idea della completezza di questa sezione basta pensare che le sono dedicate più di 160 pagine.

Nel secondo capitolo riprende la programmazione, con la descrizione di operatori, costanti, variabili ed espressioni del Basic.

Il terzo ed il quarto capitolo sono dedicati alla programmazione della grafica e dei suoni. Oltre ad un elenco completo dei comandi relativi a tali sezioni, vengono visti i modi di programmazione della grafica e del suono, anche attraverso numerosi programmi di esempio.

Viene descritta, tra l'altro, la creazione di semplici figure tramite istruzioni grafiche, ed è inclusa una completa sezione riguardante gli sprites, con la descrizione dello "sprite designer" del C/128.

Nella programmazione del suono vengono esaminate le istruzioni dedicate, con tutti i parametri che è possibile introdure, quali forma d'onda, l'ADSR, i filtri, eccetera.

Il quinto capitolo parla delle periferiche collegabili, con particolare attenzione al registratore a cassette, dato che è questa la memoria di massa più diffusa tra gli utilizzatori di home computer.

I file, la loro creazione e la relativa gestione, vengono studiati attraverso numerosi programmi di esempio. Un'altra sezione è dedicata al trattamento dei file su disco, alla directory e ad una lista di comandi del disco, completa di descrizione ed esempi di utilizzo.

Il sesto capitolo è, forse, il più interessante dato che parla del LM e del SO.

Qui vengono esaminati i concetti di memoria e di sistemi numerici (esadecimale/binario) e relativi esempi di conversione.

Viene approfondita la conoscenza del microprocessore, studiandone le funzioni svolte, anche attraverso l'Assembler del noto 6502.

In un'altra sezione è descritto il MONITOR LM incorporato e, addirittura, è presentata una lista di tutte le principali routine del SO del nuovo computer. Di ogni routine viene data la completa descrizione dell'indirizzo di partenza, del modo di utilizzo e dei registri IN/OUT.

Completano il volume le appendici, nelle quali, oltre ad una spiegazione molto dettagliata di tutti gli errori possibili, è presente anche la lista dei comandi del Basic 7.0, con abbreviazioni e codici token di ognuno di essi.

Guida di riferimento al C/128
Pagine 400
Lire 28000
Edizioni EVM Computers

Il Sistema Operativo del Cbm/64

Su questo libro non c'è molto da dire dato che rappresenta il disassemblato commentato di tutto il SO e l'interprete Basic del C/64, completo di relativo cross reference.

L'utilità di questo libro è facilmente immaginabile, in quanto si potranno conoscere tutti i segreti ed i trucchi usati dai progettisti della Commodore senza essere costretti a stampare chilometri di listato disassemblato con la propria stampante.

Inoltre è utilissimo per costruire programmi tipo Extended Basic, o altre cose del genere, utilizzando nel modo corretto le numerose routine del SO senza essere costretti a riscriverle.

Pensate alla comodità di avere 16K di ROM sotto i propri occhi per trovare immediatamente il "pezzo" necessario senza effettuare una faticosa ricerca all'interno della memoria del computer.

I commenti (in inglese, ma perfettamente comprensibili) aiutano enormemente la lettura del disassemblato che,

comunque, richiede una buona conoscenza dell'Assembler del 6502/10.

Completano il volume la descrizione delle pagine 0-4 ed una lunga tabella con gli indirizzi di accesso alle principali routine del SO non solo del C/64, ma anche del Vic 20 e dei computer della serie Cbm 3000/4000.

Il sistema operativo del Cbm C/64
Pagine 270
Edizioni EVM Computers

I segreti del 1541

Questo libro è dedicato agli utilizzatori non principianti del drive dato che non tratta solo di gestione di file e di comandi del disco, ma è anche un valido aiuto per la programmazione avanzata dell'unità a dischi 1541.

Esaminiamolo in dettaglio.

Il primo capitolo è un'introduzione che sottolinea le differenze di trattamento di file su cassetta e su disco; spiega anche in modo chiaro i vari tipi di file (sequenziali, relativi, eccetera).

Il capitolo N.2 descrive i vari comandi del drive, quali formattazione, cancellazione, inizializzazione, ed altri. Ogni comando è descritto nei minimi dettagli, quali la funzione, l'effetto, i possibili errori, ed anche vari esempi pratici di utilizzo.

Nel terzo capitolo si insegna a caricare programmi in LM, a trovare l'indirizzo di partenza dei medesimi, a registrarli sotto forma di file e a costruire caricatori Basic.

Il capitolo quattro parla della gestione avanzata dei file sequenziali, con numerosi programmi di creazione, correzione, ricerca e ordinamento degli stessi.

Nel quinto si entra nel vivo dell'argomento principale, dato che viene descritta in dettaglio la struttura di un dischetto formattato con il 1541. Si prende in esame la struttura interna della BAM e della directory, gli identificatori, si vede la differenza (interna alla directory) dei file PRG, SEQ, USR, REL, DEL, si impara a conoscere il formato dei file su disco e tante altre cose utili alla buona programmazione del 1541. Chiude il capitolo una completa descrizione sull'organizzazione dei file relativi.

Il capitolo sei è una continuazione del precedente: descrive come accedere direttamente ad un qualsiasi punto del dischetto ed esamina i buffer interni del 1541, il loro utilizzo, come leggerli, scriverli e, insomma, gestirli.

Il seguito del capitolo tratta gli altri comandi del 1541, quali Block-Read, Block-Write ed altri, con una particolare attenzione ai comandi Memory (M-R, M-W) ed USER. Il tutto è contornato da numerosi programmi ed esempi di utilizzo.

Il volume prosegue con la descrizione delle prime otto pagine di memoria (da $0000 a $07FF) e del disassemblato commentato dei 16K del SO del disco.

Grazie a manuali di questo tipo è possibile scrivere programmi come Turbo-Disk, Fast-Copy, ed altri che richiedano una specifica conoscenza del SO del disco. Per il corretto utilizzo di questa parte del libro è necessaria, ovviamente, una buona conoscenza dell'Assembler del 6502.

In appendice sono riportati utili programmi per la gestione del disco, tra cui un disk-monitor interamente in LM e quindi velocissimo. Una dettagliata descrizione degli errori conclude gli argomenti del libro.

Insieme col manuale viene fornito lo schema elettrico del drive (utile in caso di guasti). Purtroppo tale schema non è molto chiaro, dato che tutte le sigle e i nomi sono stati scritti a mano ed in alcuni punti appaiono confusi e di difficile comprensione.

Con una cedola presente nel libro, la EVM pone condizioni vantaggiose per l'acquisto del disco contenente i programmi riportati nel volume ed anche alcuni manuali tra i quali "Corso di Assembler" e "Le periferiche Commodore".

I segreti del 1541
Pagine 290
Edizioni EVM Computers

SCAMBIATEVI LE LISTE

Vendo corso di basic "Conoscere il computer direttamente dal computer" completo, 30 giochi, 15 programmi a lire 50.000. *(Alberto Vico, via Sant'Antonio da Padova 2 - 10121 Torino. Tel. 011-538919).*

Vendo per Vic20 espansione 3/8/16 K per C64 cedo interessante cartuccia Isepic. *(Gianni Cottogni, via Strambino 23 - 10010 Carrone (Torino). Tel. 0125-712311).*

Vendo modem C64 300/600/1200 baud autodial compatibile agli standard europei ed americani più software su disco istruzioni, il tutto a lire 200.000. *(Stefano Malaisi, via Marsala 31 - 40126 Bologna. Tel. 051-235492).*

Vendo cartuccia Vic 20 e introduzione al basic I/II sempre Vic20. *(Luca Spangaro, via Liguria 21 - 33100 Udine. Tel. 0432-471707).*

Vendo 2 paddle, duplicatore di cassette RCP per Commodore 64. *(Cristiano cravanzola, via Prof. Oliva 16 - 12011 Borgo S.D. Tel.0171-769659).*

Cerco routines per stampare copia video Hi-Res con la stampante Commodore MCS-801. *(Piero Nobile, via Martiri 223 - 18038 Sanremo. Tel. 0184-74941).*

Vendo in blocco Commodore 64, drive 1541, datasette, stampante Mps 803 e circa 400 dischetti. *(Enzo Iovino, via del taccaro 30 - 84012 Angri).*

Vendo C64, drive 1541 - Mps 802, monitor fosfori verdi inperfette condizioni più vari programmi. *(Ruggero Rigoldi, via Zanella 7 - 20052 Monza. Tel. 039-836620).*

Vendo C64 più registratore e/o Vic20 più registratore il primo a lire 400.000 e il secondo a lire 150.000. *(Adelchi Crespan, via Toniolo 35 - 31015 Conegliano (Tv). Tel 0438-63318).*

Vendo Commodore Plus/4 più registratore, in omaggio giochi utilities. *(Paolo Borghetti, via Cresseri 25 - 25020 Pralboino (Bs). Tel. 030-954211).*

Cerco Modem 300 baud max lire 200.000. *(Massimiliano Di Lorenzo, piazza 185mo Reg. Folgore 16 - 57100 Livorno. Tel. 0586-580796).*

Vendo C64, registratore C2N 1530, joystick "The boss", tasto reset, circa 170 programmi. *(Francesco Ranzato , via Roma 62 - 35010 Vigodarzere (Pd). Tel. 049-702651).*

VendoC64 più stampante grafica bidirezionale *Mps 803 più programmi. (Paolo Cerabolini, via Brioschi 22 - 20136 Milano. Tel. 02-8371 945).*

Vendo per C64 cartuccia nuova "Turbo disk" con manuale in italiano più libro commodore 128. A lire 60.000 o separatamente a lire 40.000 e 20.000 rispettivamente. *(Federico Solla, via G. Piazzi 39†D - 80137 Napoli. Tel. 081-442428).*

Vendo Atari VCS 2600 più joystick più paddles più cartucce, alimentatore a lire 120.000. *(Fabio Calabrese, via Spagna 33 - 90146 Palermo. Tel. 091-502509).*

Cerco utenti modem 300 baud (standard europeo) per collegamenti. *(Marco Cortesi, via Libertà 13 - 47042 Cesenatico. Tel. 0547-80961).*

Compro hardware, computer a prezzi bassi. Anche malridotti. *(Christian Mauro, via Cassio P.se 4-2 - 43100 Parma. Tel. 05221-42771).*

Vendo ZX Spectrum, monitor tv e relativi cavetti, registratore Sharp Ce-152, interfaccia joystick Ram turbo più corso completo video basic, diversi programmi e libri. Disponibile vendita singoli prezzi. *(Giovanni Pilo, via Palabanda 23 - 09123 Cagliari. Tel. 070-656872).*

Cerco drive 1541 cerco stampante Commodore o affine. *(Gaetano Santoro, via Elia Crisafulli 30 - 90128 Palermo. Tel. 091-427914).*

Vendo Plotter Commodore 1520 a lire *250.000, floppy disk a lire 350.000. (Rosario Di Betta, via Parodi 42 - 88019 Vibo Marina (Cz). Tel. 0963-240834).*

Cerco *Turbo tape per C16. (Marco Cammarata, via Claudia Augusta 20 - 39100 Bolzano).*

SCAMBIATEVI LE LISTE

Andrea Casale - Via delle Azalee 27 - 00048 Nettuno - Tel. 06/9804478

Federico Fracassi - Lungarno Tripoli 7/A - 56029 Santa croce S/A (Pi) - Tel. 0571/34640

Davide Valloggia - Via Giuseppe Verdi 101 - 28021 Borgomanero (No) - Tel. 0322/82692 - 844822

Marco Benvenuti - Via C. Mazza 43 - 44100 Ferarra - Tel. 0532/901976

Antonio Irpino - Visa Nazionale 114/B - 83013 Mercogliano (Av) - Tel. 0825/682258

Andrea Trabacchi - Via Pordenone 22 29100 Parma - Tel. 0523/62446

Mauro Serafini - Via Renzi 9 - 47037 Rimini - Tel. 0541/54652

Guido Scalmato - Via Trilussa 37 - 04011 Aprilia - Tel. 06/923447

Dario Giannessi - Via Cariola 3 - 56100 Pisa - Tel. 43817

Sebastiano Suraci - Via Pallanza 8 - 33100 Udine - Tel. 0432/45274

Fabio Olgiati - Via Puecher 6 - 20020 Dairago - Tel. 0331/431494

Antonino Baglione - Via Castellana 318 - 90135 Palermo

Gianni Todero - Via Vecchia Trevigiana 80 - 31015 Conegliano - Tel. 0438/60314

Tamara Cacciamani - Località Il Piano - 58017 Pitigliano (Gr) - Tel. 0564/615622

Angelo Orlandi - Via Delle Albizzie 40 - 00942 Roma - Tel. 06/288368

Leonardo Gambini - Via Pisana Traversa dei Bocci 36 - 55100 San Donato (Lu) - Tel. 0583/510562

Eugenio Baioni - Piazza Cavalieri Vittorio Veneto 4 - 04019 Terracina (Lt) - Tel. 0773/724252

Massimiliano Camuffo - Via Bissa 44 - 30173 Mestre (Ve) - 041/986707

Paolo Nannotti - Via Grosseto 26 - 53049 Torrita (Si) - Tel. 0577/687032

Marco Coccia - c/o Loredana Ammazzalorso - Via Sangro - 65010 S Santa Teresa di Spoltore (Pc) - Tel. 085/209630

Mario Criscuolo - Piazza B. Crivelli 31 00159 Roma - Tel. 06/4390265

Massimo Musacchi - Via Aminta 11 - 44038 Ferrara - Tel.0532/464110

Carlo Ballardin - Via San Pietro 59 - 35100 Padova - tel. 049/20272

Rosario Bonavita - Via F. della Monica 60 - 84100 Salerno - Tel. 089/356832

Carmelo Petraglia - Via Max Casaburi 8 - 84100 Salerno - Tel. 089/236873

Bruno Mialich - Via Cardinal Jacopo Monico 23 - 30174 Mestre (Ve) - Tel. 041/914301

Florindo Costa - Pordenone - Tel. 0434/652500

Daniele Petracca - Via Carlo Pisacane 26 - 04100 Latina - Tel. 0773/497071

OPERAZIONE FEDELTÀ SYSTEMS.

I VINCITORI!

Pubblichiamo l'elenco dei quaranta lettori che sono risultati vincenti alla prima estrazione mensile del concorso "Vince chi legge". L'estrazione è stata effettuata il 28 marzo 1986 presso la sede della Systems, alla presenza di un funzionario dell'Amministrazione finanziaria.

I primi cinque estratti vincono una stampante Commodore a margherita; dal sesto al decimo estratto riceveranno una printer-plotter Commodore; dall'undicesimo al quarantesimo un libro a scelta della biblioteca Systems oppure un abbonamento a una rivista Systems. Tutti i coupons ricevuti concorreranno all'estrazione del premio finale, consistente in una moto Cagiva Electra 125cc.

Ed ecco i primi quaranta nomi (tra parentesi il numero d'ordine del coupon di partecipazione):

1 Zizzi Giuseppe - Martina Franca (TA) - (598)
2 Carulli Luigi - Monte S. Angelo (FG) - (125)
3 Beltrami Massimo - Ravenna - (770)
4 Stedile Luigino - Ferragnolo (TN) - (663)
5 Fumia Bruno - Torino - (257)

6 Galeazzi Gabriele - Carpi (MO) - (419)
7 Marchesiesich Giovanni - Trieste - (199)
8 Tomasella Miguela - Zoppè S. Vendemiamo (TV) - (388)
9 Curci Enrico - Bari - (550)
10 Totaro Atonio - Latina - (271)

11 Prosperi Alessio - Empoli (FI) - (454)
12 Perrella Antonio - Cast. Mare Stabia - (NA) - (744)
13 Gaudiosi Rodolfo - Capaccio Scalo (SA) - (73)
14 Marciano Antonio - Piano di Sorrento (NA) - (395)
15 Luciani Vincenzo - Benevento (NA) - (166)
16 Bellini Francesco - Firenze - (161)
17 Sintucci Guido - Cesena (FO) - (578)
18 Carollo Paolo - Este (PD) - (590)
19 Losi Umberto - Cuglieri (OR) - (767)
18 Farè Silvia - Milano - (707)
21 Sorrentino Giuseppe - Salerno - (343)
22 Stompanato Gaetano - Acerra (NA) - (479)
23 Abenavoli Renato - Roma - (688)
24 Colombo Gilberto - Costamagnaga (CO) - (626)
25 Silvestri Raffaele - Napoli - (313)
26 Leto Giuseppe - Torino - (558)
27 Del Grande Mirko - Pesaro - (559)
28 Tumminia Stefano - Reggio C. - (508)
29 Ferruzzi Claudio - Torino - (57)
30 Conte Antonio - Andria (BA) - (30)
31 Tugnoli Francesco - Bologna - (376)
32 Pacchiarini Loris - Portoferrario (LI) - (110)
33 Gentili Liberato - Ascoli Piceno - (786)
34 Pozzoli Dante - Narni (TR) - (564)
35 Lucchini Giuseppe - Lido di Roma - (180)
36 Trofa Biagio - S. Giorgio del Sannio (BN) - (111)
37 Russo Gaetano - Quarto (NA) - (404)
38 Merolla Paolo - Napoli - (108)
39 Maxia Renato - Cagliaci - (287)
40 Capo Stefano - Roma - (54)

INVIARE TUTTA LA PAGINA ANCHE SE SI UTILIZZA UNA SOLA SCHEDA

Nome .... Cognome ....

Via .... N° .... CAP. .... Città ....

Telefono .... Orario ....

Registrate il mio abbonamento annuale a Commodore Computer Club.

☐ Ho versato oggi stesso il canone di L. 35.000 a mezzo c/c postale n° 37952207 intestato a:
Systems Editoriale - V.le Famagosta, 75 - 20142 Milano

☐ Ho inviato oggi stesso assegno bancario n..
per l'importo di L. 35.000 intestato a Systems Editoriale

Si prega di scrivere il proprio nome e l'indirizzo completo in modo chiaro e leggibile. Inviare la fotocopia del bollettino di c/c postale.

Considerando che i numeri 1, 2 e 7 sono esauriti, vogliate inviarmi i numeri arretrati .... al prezzo di L. 5.000 cadauno per richieste fino a 4 numeri, o di L. 4.000 cadauno per richieste oltre i 4 numeri arretrati, e perciò per un totale di L............... Sono a conoscenza che i fascioli suddetti non saranno inviati in contrassegno e, pertanto, ho provveduto oggi stesso a versare il canone di L.................. a mezzo c/c postale n. 37952207 intestato a:
Systems Editoriale - V.le Famagosta, 75 - 20142 Milano

STATISTICA

Non posseggo un computer ☐

Posseggo un C64	si ☐	... no ☐
Posseggo un VIC 20	si ☐	... no ☐
Posseggo un Commodore Plus 14	si ☐	... no ☐
Posseggo un Commodore Plus 16	si ☐	... no ☐
Posseggo un registratore dedicato	si ☐	... no ☐
Posseggo un drive 1541	si ☐	... no ☐
Posseggo una stampante	si ☐	... no ☐
Posseggo un monitor	si ☐	... no ☐

COLLABORAZIONE

A titolo di prova vi invio un articolo e la cassetta........................disco..................................
col programma che intendo proporre per la pubblicazione di cui garantisco l'originalità.

DOMANDA/RISPOSTA ....

....

....

....

....

....

RICHIESTA ARGOMENTI

Mi farebbe piacere che Commodore Computer Club parlasse più spesso dei seguenti argomenti:

1/ ....

2/ ....

3/ ....

4/ ....

GIUDIZIO SUI PROGRAMMI DI QUESTO NUMERO

Ho assegnato un voto da 0 a 10 ai programmi che indico di seguito:

A/ .... Voto ....

B/ .... Voto ....

C/ .... Voto ....

D/ .... Voto ....

PICCOLI ANNUNCI

....

CERCO/OFFRO CONSULENZA

....

INVIARE IN BUSTA CHIUSA E AFFRANCANDO SECONDO LE TARIFFE VIGENTI A:

COMMODORE COMPUTER CLUB

V.le Famagosta, 75
20142 Milano

INVIARE TUTTA LA PAGINA ANCHE SE SI UTILIZZA UNA SOLA SCHEDA

Nome ______ Cognome ______

Via ______ N° ______ CAP. ______ Città ______

Telefono ______ Orario ______